



Servojärjestelmän simulointi digitaalisen kaksosen avulla Matlab- ja Siemens NX - ympäristössä

Max Caroe

OPINNÄYTETYÖ
Toukokuu 2020

Konetekniikka
Koneautomaatio

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Konetekniikka
Koneautomaatio

CAROE, MAX:

Servojärjestelmän simulointi digitaalisen kaksosen avulla Matlab- ja Siemens NX -ympäristössä

Opinnäytetyö 60 sivua, joista liitteitä 1 sivu
Toukokuu 2020

Työn tavoitteena oli tutkia ja mallintaa simuloitava kokoonpano koululla olevasta fyysisestä laitteesta. Tämä opinnäytetyö tehtiin toimeksiantona Tampereen ammattikorkeakoululle. Simulointi tapahtuu Matlab Simulink -ohjelman avulla yhdessä Siemens NX-suunnitteluohjelman kanssa. Työn on määrä tulla osaksi koulun oppimisympäristöä ja sen avulla voidaan tutustua kaksosen luomisprosessiin sekä simuloitavan mallin toimintaan.

Opinnäytetyössä käsitellään digitaalisen kaksosen teoriaa, sen luomisprosessia, yhteyden muodostamista käytössä olevien ohjelmien välille sekä pyritään tuomaan esille ongelmia, joita mallin luomisen aikana kohdattiin. Lisäksi tutustutaan lyhyesti digitaalisen kaksosen simulointiin pelkästään Matlab-ohjelmassa Simscape Multibody -ympäristössä.

Tuloksena syntyi simuloitava malli, jonka avulla voidaan konkreettisesti havainnollistaa servojärjestelmän PID-säätöä. Malli toimii myös valmiina pohjana muihin simuloitaviin säätömenetelmiin.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Mechanical engineering
Machine automation

CAROE MAX:

Simulation of a servosystem with a digital twin in Matlab- and Siemens NX -environment

Bachelor's thesis 60 pages, appendices 1 page
May 2020

The thesis was carried out for Tampere University of Applied Sciences (TAMK). Its purpose was to research and design a simulation model of a device located in the university's automation laboratory. The simulation is taking place in Matlab Simulink together with Siemens NX 3D design software. This thesis and simulation are to become part of a learning environment in use at the university. With it students can introduce themselves to the digital twins creation process and its model.

This thesis provides a discussion on the digital twin and its creation, connecting of Matlab Simulink to Siemens NX together and insight of the problems and challenges faced during this process. Additionally, a brief overview is given on the simulation of a digital twin solely in the Simscape Multibody environment of Matlab.

As a result of this study a simulation model was created, which enables concrete demonstration of a PID-controlled servosystem. The model is also fit to act as a starting point for future development and projects.

Key words: digital twin, simulation, servo control system

SISÄLLYS

1	JOHDANTO	7
1.1	Työn tausta	7
1.2	Työn tavoitteet.....	8
2	TEORIA	9
2.1	Digitaalinen kaksonen	9
2.2	Suunnittelu	11
2.3	Servomoottorijärjestelmä	12
2.4	PID	13
2.5	Työssä käytetyt sovellukset	15
3	DIGITAALISEN KAKSOSEN LUOMINEN.....	18
3.1	Siemens NX assembly	19
3.2	Simcenter Motion	20
3.2.1	Simulointiympäristö	21
3.2.2	Linkit	23
3.2.3	Liikeradat.....	25
3.2.4	Liikeratojen rajoittaminen.....	26
3.2.5	Jouset.....	27
3.2.6	Sensorit	28
3.2.7	I/O.....	30
3.2.8	Ratkaisumalli	32
3.2.9	Kitka	32
4	SIMULINK.....	34
4.1	PID säätö	35
4.2	Värähtelyn huomioonottaminen PID-säädössä	36
4.3	Moottorin mallinnus	37
4.4	Simulointi	38
5	SIMSCAPE MULTIBODY ENVIRONMENT	43
5.1	Asennus	43
5.2	Mallin tuominen Multibody-ympäristöön	45
5.3	Simuloiminen.....	48
6	TULOKSET	50
6.1	3D-malli	50
6.2	Simulaatio	51
6.2.1	Pulssiohjaus	52
6.2.2	PID-säätö ilman värähtelyn kompensointia	54

6.2.3 PID-säätö värähtelyn kompensoinnin kanssa.....	55
7 POHDINTA	57
LÄHTEET.....	58
LIITTEET	60
Liite 1. Matlab Simulink-malli.	60

ERITYISSANASTO

I/O	Input ja Output
MCD	Mechanical concept designer
NX	Siemensin kehittämä 3D-suunnitteluohjelmisto
PLM	Product life cycle management
PID	Proportional-integral-derivative
PWM	Pulse width modulation

1 JOHDANTO

Digitaalinen kaksonen on käsitteenä tuore, vaikka sen luomiseen olevat työkalut ovat olleet markkinoilla käsitettään pidempään. Tiivistettynä digitaalinen kaksonen on eri teknologioiden integroiminen kokonaisuudeksi, joka tuottaa useita funktioita (Sustainability, 12, 1088; doi:10.3390/su12031088). Ohjelmat, jotka on suunniteltu muun muassa tietokoneavusteiseen mallintamiseen, tuotteen elinkaarenhallintaan tai elementtimenetelmien (*finite element analysis*) laskentaan ovat olleet olemassa pidemmän aikaan. Teknologioiden integroimisessa yhdeksi kokonaisuudeksi on kyse juuri tämänkaltaisten ohjelmien toiminnallisuuksien yhdistämisestä ja sellaisten tulosten tuottamisesta, joita ei yksinään tällaisilla ohjelmille pystyittäisi tuottamaan. Tässä opinnäytetyössä yhdistettiin tietokoneavusteinen suunnittelu ja numeerinen laskenta toimimaan yhdessä suunnittelun näkökulmasta.

1.1 Työn tausta

Opinnäytetyö tehtiin toimeksiantona Tampereen ammattikorkeakoululle. Korkeakoulun automaatiolaboratorion tiloissa sijaitsevasta laitteesta tehtiin 3D-malli Siemens NX-suunnitteluohjelmalla sekä mallia simuloitiin Matlab-sovelluksella. Laitteessa on kaksi servomoottorien avulla ajettavaa ruuviakselia, joiden työkalupään toimintaa simuloitiin eri käyttöympäristöissä. Laite on ripustettuna jousien varassa, ja joidenka aiheuttamaa värähtelyä kompensoitiin PID-säätimen avulla. Mallin on tarkoitus toimia suunnittelevasta ja analysoivasta näkökulmasta, minkä avulla voidaan tutustua laitteen toimintaa sekä käyttää laitetta erilaisissa olosuhteissa.

1.2 Työn tavoitteet

Työn päätavoitteena oli luoda laitteesta tarkka 3D-malli ja simulaation mallin toiminnasta. Työn osatavoitteita oli seuraavia:

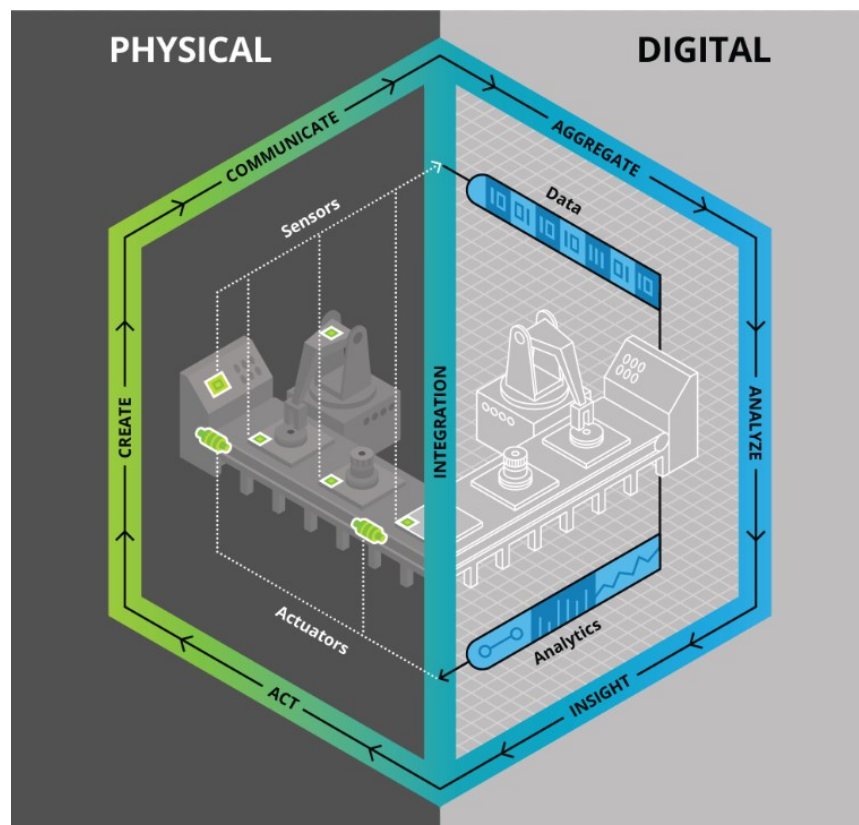
- Kaikkien laitteen komponenttien mallintaminen 3D-malleiksi
- Kokoonpanon luominen
- Simulaatiomallin luominen Siemens NX-ohjelmassa
- Simulaatiomallin valmistelu Matlab-yhteissimulaatiota varten
- Yhteyden muodostaminen ohjelmien välille
- Simulink-mallin luominen
- Tulosten johdonmukaisuus eri mallien välillä
- Moottorien mallinnus Simulink-ympäristössä
- Simscape Multibody-ympäristöön tutustuminen
- Tulosten tarkastelu

Työn tavoitteet rajattiin simulaatiomallin luomiseen. Tällöin mallin identifiointi laitteeseen jätettiin tavoitteiden ulkopuolelle. Siemens NX- ja Matlab-ympäristöissä tapahtuvan simulaation lisäksi haluttiin tutustua Matlabin Simscape Multibody-ympäristössä suoritettavaan digitaalisen kaksosen luomisprosessiin.

2 TEORIA

2.1 Digitaalinen kaksonen

Digitaalisella kaksosella tarkoitetaan lähtökohtaisesti tietokoneella tehtyä mallia fyysisestä laitteesta tai prosessista. Siinä yhdistyy laitteen sekä ympäristön fyysiset ominaisuudet simuloitavaksi kokonaisuudeksi. Digitaalinen kaksonen voi olla joko yksittäinen tarkasteltava prosessin osa tai kokonainen tuotantoketju. Kaksonen voi olla myös pilvipohjainen reaaliajassa päivittyvä prosessikartta ilman sen kummempaa fyysisten komponenttien kuvausta. Tässä opinnäytetyössä keskitytään yksittäisen laitteen toiminnan mallintamiseen.

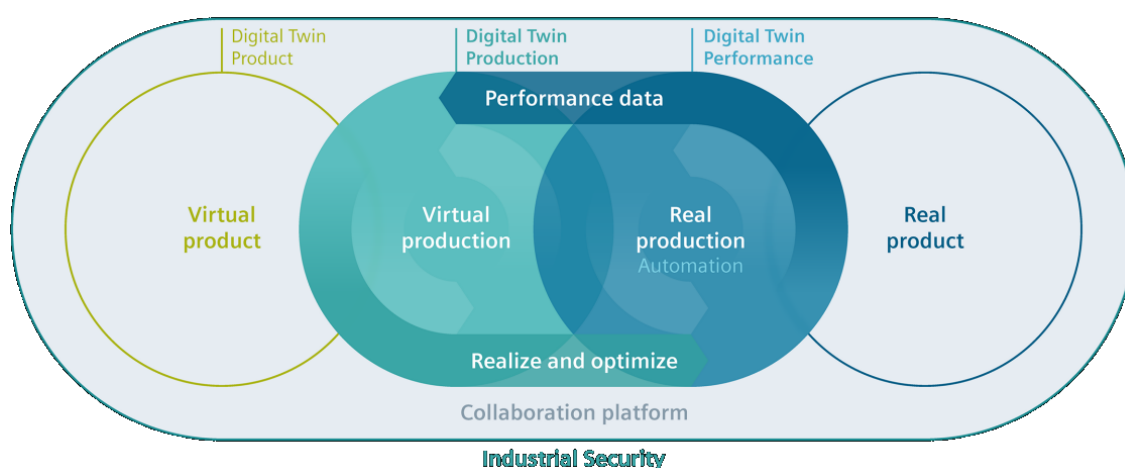


Kuva 1. Digitaalisen kaksosen kiinnittyminen fyysiseen ympäristöön. (<https://i.pinimg.com>)

Prosessin simulointi tietokoneella on usein kustannustehokkaampaa kehitystyön kannalta. Laitteen ensimmäistä prototyyppiä ei välttämättä tarvitse valmistaa fyysisesti ollenkaan, ja kun ensimmäinen prototyyppi päätetään valmistaa, on siitä pystytty tunnistamaan ja korjaamaan suurimmat virheet.

Muutosten seurauksien ennustaminen ja jo olemassa olevien ongelmien korjaaminen ilman digitaalista sisarustaan voi olla haastavaa ja joissain tapauksissa myös vaarallista. Suurten kokonaisuuksien ja prosessien muutokset vaativat perustavanlaatuista selvitystyötä eikä virheisiin välttämättä ole varaa. Tällaisten tarkkaan harkittujen muutosten seurausten ennustaminen digitaalisen kaksohen avulla ei aiheuta laiterikkoja, pysäytä tuotantolinjaa tai vaaranna ihmisiä. Erityisesti suunnittelussa digitaalinen kaksohen luodaan ennen fyysistä vastakappaletaan, mutta on myös mielekästä tarkastella jo olemassa olevaa prosessia tai laitetta digitaalisen kaksoiskappaleen avulla.

Digitaalisen kaksohen tuomat hyödyt eivät ole kuitenkaan aina helposti saavutettavissa. Mallinnus ja simulointi tuovat omat haasteensa omine rajoituksineen eivätkä nämä rajoitukset ole välttämättä suoraviivaisesti ratkaistavissa. Nämä rajoitukset riippuvat prosessin tarkkuudesta ja siitä, kuinka tarkasti ne halutaan kuvata tietokonesimulaatiossa. Reunaehtojen määrittäminen digitaalisen kaksohen suunnittelussa on olennaista, sillä vaikka mallinnukseen käytettäisiin merkittäviä resursseja, ei digitaalinen kaksohen välttämättä koskaan tule kuvastamaan tuotteen tai prosessin toimintaa täydellisesti. Viimekädessä toiminnan hienosäätö tapahtuu itse fyysiseen oikean elämän prosessiin.



Kuva 2. Collaboration platform. (<https://assets.new.siemens.com>)

2.2 Suunnittelu

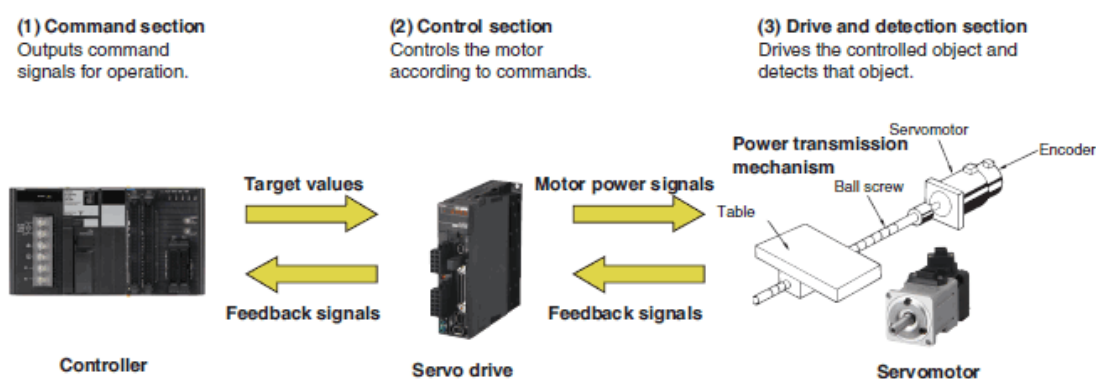
Digitaalisen kaksosen mallintamista lähestytään suunnittelusta asti sen käyttötarkoituksensa mukaan. Ennen varsinaista mallinnusta on siis määriteltävä mitä on tarkoitus mallintaa ja mitä siltä odotetaan. Digitaalisen kaksosen malli tukee suunnittelua, tuotantoa tai suorituskykyperäistä suunnittelua, mutta ei pysty täysivaltaisesti tukemaan näitä kaikkia samaan aikaan (Ridal M. 2019). Suunnittelussa mallille määritellään reunaehdot, joiden avulla voidaan rajata mallin toiminnallisuus oman käyttötarkoituksensa sisälle. Malleja voidaan luoda samasta prosessista useampia, jolloin mallien on mahdollista myös keskustella keskenään.

Suunnittelussa reunaehtojen määrittäminen on tärkeää myös, jotta osataan tunnistaa prosessille tärkeät osat. Mallinnuksen ja simuloinnin eri prosessien osien tarkkuus voi vaihdella niiden toiminnan mukaan. Vähäpätöisimmille prosessien osille voi riittää paljon epätarkempi kuvaus sen toiminnasta simulaatiossa. Puolestaan halutaan keskittää resursseja tärkeimpien toimintojen yksityiskohtaisempaa tarkastelua varten.

Digitaaliseen mallintamiseen ja kaksosen luomiseen on olemassa useita erilaisia sovelluksia. Mallin käyttötarkoitus on omiaan rajaamaan käytettävissä olevien sovellusten määrää tarpeiden mukaan. Esimerkiksi Siemens NX -suunnitteluympäristö tarjoaa muun muassa mahdollisuuden staattisten kuormien tarkasteluun FEM-analyysin avulla simulaation aikana. Mallinnusta ja sen käyttötarkoitusta on hyvä tarkastella myös kustannusnäkökulmasta. Useat ohjelmat ovat maksullisia tarjoten vuosittaisia tai kuukausittaisia lisenssejä niiden käyttöön. Tähän on hyvä varautua digitaalisen kaksosen suunnittelun kustannuksia arvioitaessa. On myös pohdittava tuottaako digitaalinen kaksonen sellaista lisäarvoa suunnittelulle ja tuotannolle, mikä on nähtävissä lopputuotteen turvallisuudessa, arvossa tai laadussa.

2.3 Servomootorijärjestelmä

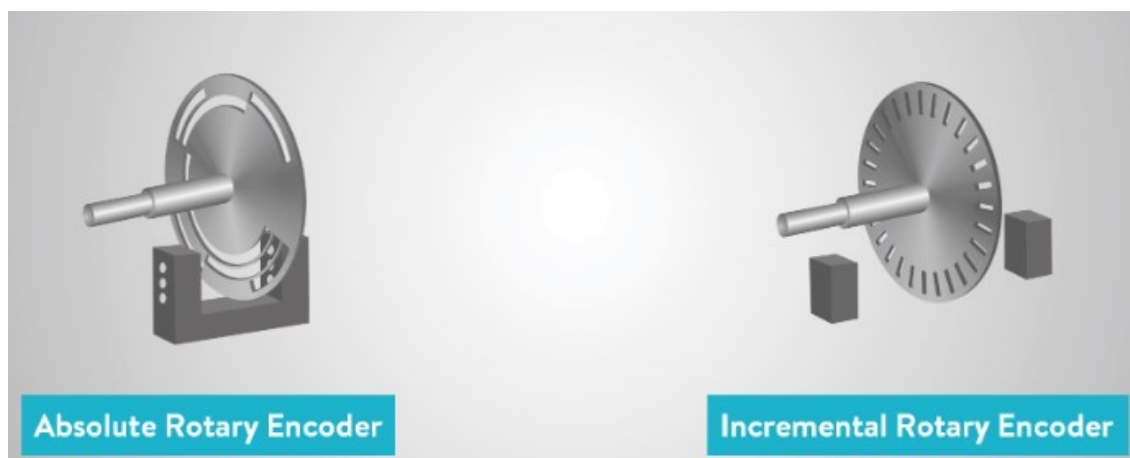
Servokäytöllä tarkoitetaan yleisimmin takaisinkytkettyä toimilaitetta. Toimilaitteesta riippuen se voi toimia joko sähköisesti, hydraulisesti tai pneumaattisesti. Tässä työssä painotutaan sähköisen servomoottorin toimintaan. Takaisinkytkennällä tarkoitetaan anturia, jonka avulla moottorin paikkaa, kierroksia tai nopeutta voidaan mitata. Servomoottorin säätö tapahtuu tämän anturitiedon puitteissa. Usein servomootoreissa on tämän lisäksi sisäänrakennettu jarru, jonka tarkoituksena on avustaa servomoottorin paikoituksessa, asetetun paikan ylläpitämisessä sekä toimia hätätilanteissa pysäyttävänä voimana. Jarru on erillään moottorin virtapiiristä ja on tyypillisesti rakennettu niin, että jännitteen katketessa jarru aktivoituu. (Technical explanation for servomotors, Omron guide).



Kuva 3. Servojärjestelmän komponentit. (<https://www.ia.omron.com>)

Servomoottorikäytössä moottoria ohjataan servovahvistimen avulla. Servovahvistimen tehtävänä on vastaanottaa ylemmän tason toimilaitteelta ohjaussignaali ja ohjata moottorin liikettä sen mukaan. Usein ohjaussignaali on joko jännitetieto tai virtaviesti riippuen käytettävien laitteiden vaatimuksista. Sen tehtävänä on myös tarjota moottorille sen käyttämä jännite ja virta. Servovahvistimeen kytetään myös moottorilta tuleva anturitieto, jonka puitteissa moottoria käytetään. Servovahvistimen sisällä näiden kahden paikkatiedon, ohjaussignaalin ja anturitiedon, vertailu tapahtuu. Tavallisimmin säätöön käytetään PID-säätöä, mutta muutkin säätötavat ovat mahdollisia. (FA Equipment for beginners (Servos). 2014.)

Anturit voidaan jakaa pääpiirteittäin kahteen ryhmään: absoluuttisiin ja inkrementaalisiin. Absoluuttinen sensori pystyy mittaamaan paikkatietonsa myös virtakatkoksen aikana, jolloin servokäytössä erillistä kalibrointia ei tarvita. Anturi itsessään tietää missä se sijaitsee. Rajoituksina absoluuttisissa antureissa on niiden mitattavan alueen rajallisuus, usein sen monimutkaisempi rakenne sekä joissain malleissa myös osien kuluminen.



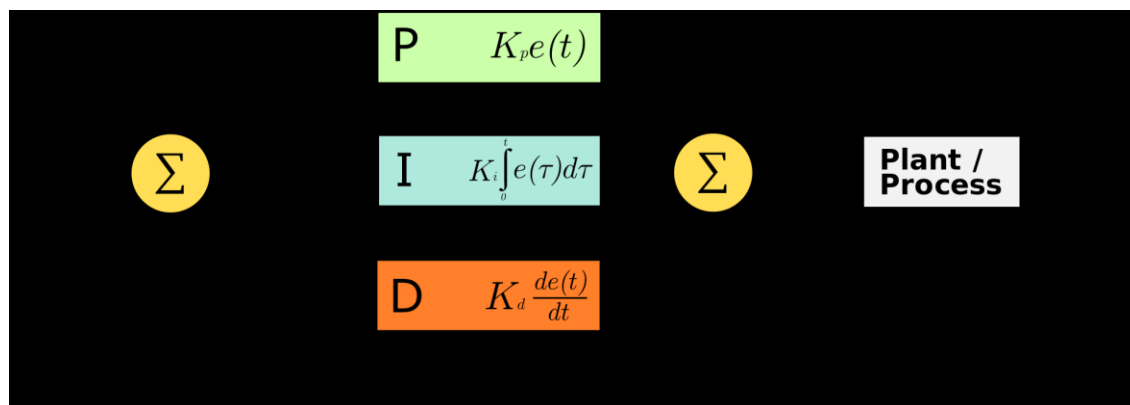
Kuva 4. Absoluuttinen ja inkrementaalinen kiertoanturi. (<https://realpars.com>)

Inkrementaaliset anturit puolestaan mittaavat liikettä ja sen nopeutta. Anturin mitaus perustuu kahteen luettavaan arvoon, joiden välillä vallitsee vaihe-ero. Vaiheeron avulla anturin signaaleista on luettavissa liikkeen suunta ja sen nopeus. Anturi tulee kalibroida muun muassa sähkökatkoksen jälkeen. Kalibroinnissa sille kerrotaan missä se sijaitsee, jonka jälkeen anturilla seuratun liikkeen määrää verrataan sille kalibroinnissa asetettuun lähtöarvoonsa. Usein moottorille on määritetty kotipaikka, jota käytetään kalibroinnin suorittamiseen. Sivistyneemmissä systeemeissä kalibroinnille on oma aliohjelmansa, joka suorittaa kotipaikan määrittämisen automaattisesti käyttäen hyväksi järjestelmään rakennettuja rajakytkimiä.

2.4 PID

Yksinkertaisimmillaan PID-säätö perustuu kolmen funktion, proportionaalien, integraalin ja derivaatan summaan. Näiden funktioiden kerrannaisilla voidaan sää-

dellä niiden voimakkuutta säätöä tehdessä. Minkä tahansa näiden kertoimien arvoiksi voidaan määritellä nolla, jolloin se ei osallistu säätöön ollenkaan. Usein puhutaan tällöin PD- tai PI-säätimestä. PID-säädin on yleisin säätötapa prosessin hallintaan. Se on helposti sovellettavissa, suhteellisen yksinkertainen asentaa ja sen säätöperiaate on helppo ymmärtää. (Bill Drury. 2001. Control techniques: Drives and controls handbook, 104.).



Kuva 5. PID-säädin prosessikuvaus. (Arturo Urquizo)

Proportionaalifunktion tehtävä on skaalata mitattavan virheen arvo. P-funktio määrittelee pitkälle prosessin nopeuden ja vasteajan, mutta käytettäessä liian suurta P-funktion arvoa prosessi saattaa oskilloida ja vielä aggressiivisemmilla arvoilla muuttua kokonaan epästabiiliksi. P-funktiota käytetään melkein poikkeuksetta yhdessä vähintään joko integraalifunktion tai derivaattafunktion kanssa. Integraalifunktion avulla voidaan kuroa stabiilin tilan virhe pois ajan kanssa. Tämän opinnäytetyön kohdalla integraalifunktiota ei käytetty lopullisessa versiossa laisinkaan. Derivaattafunktion tehtävänä on pienentää säätimen säätöarvoa, kun virhearvo kasvaa nopeasti. Derivaattafunktion arvo on suoraan verrannollinen prosessin virhearvon kasvunopeuteen. (Oguntoyinbo O. J. 2009.).

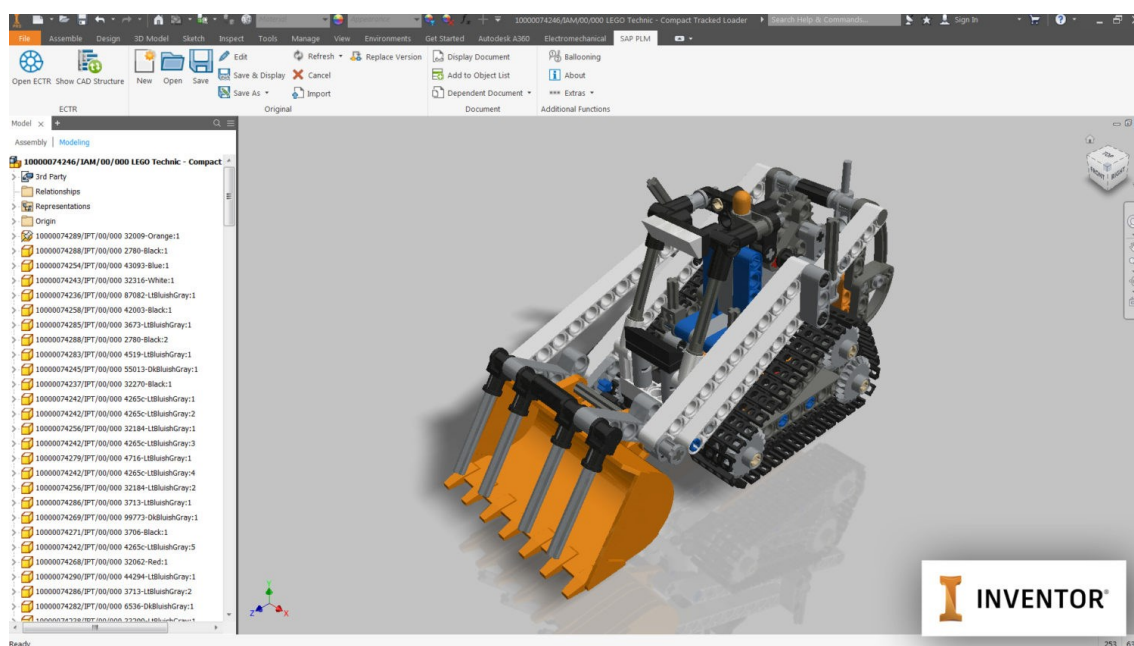
PID-säätimelle on ominaista sen kankeus, kun prosessissa on useampia muuttujia tai kun prosessin luonne muuttuu eri tilanteissa. Kun kuormitus kasvaa tai sen herkkyys säädölle muuttuu, on PID-säätö joko tehtävä uudelleen vastaamaan muuttunutta tilannetta tai käytettävä PID-säätimen kanssa dynaamista prosessiarvojen mukana muuttuvaa säädintä. Tätä kutsutaan adaptiiviseksi säädöksi (*gain scheduling*). Adaptiivisessa säädössä prosessi kartoitetaan, jolloin tavoitteena on säätää PID-säädin prosessin jokaista tilannetta kohden ja muuttaa

säätimen arvoja sen mukaan. (Kangasrääsiö A. 2010, 19.) Tämä kartoitus voidaan tehdä joko manuaalisesti tai automaattisesti.

2.5 Työssä käytetyt sovellukset

Työn kulku on jaoteltavissa kolmeen sovellukseen. Työ aloitettiin piirtämällä laitteen osat Autodesk Inventor -sovelluksella, josta jatkettiin kokoonpanon ja simulaatiomallin luonnilla Siemens NX -sovelluksella. Viimeisenä otettiin käyttöön MatLab -sovellus, jonka avulla suoritettiin simulaation laskennallinen osuus yhteiskäyttönä Siemensin sovelluksen kanssa.

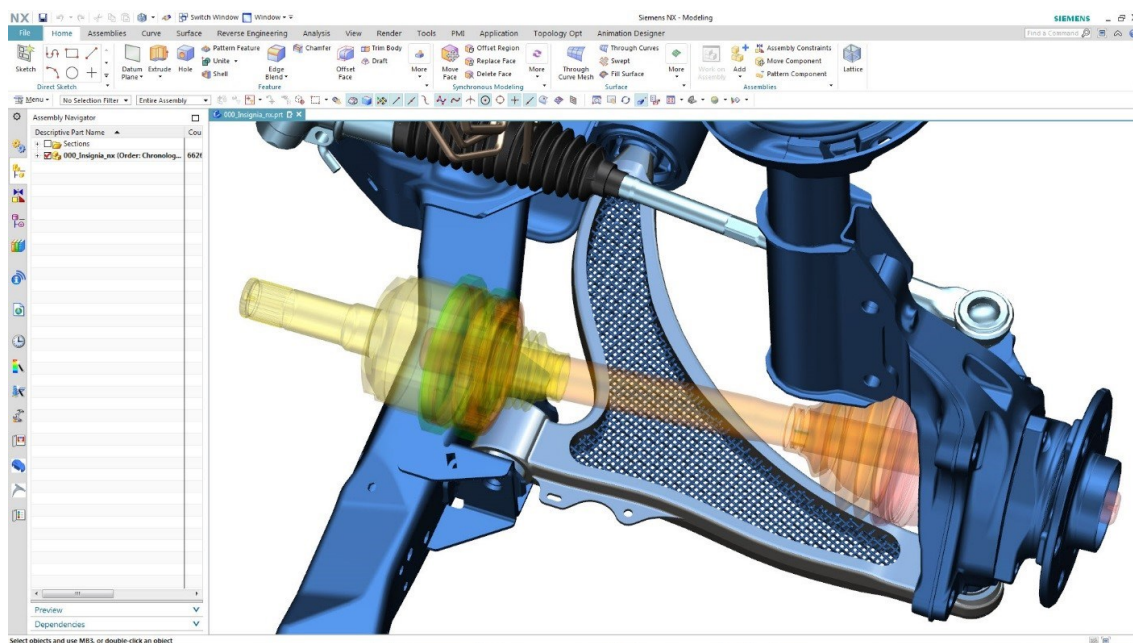
Autodesk Inventor on tietokoneavusteiseen suunnitteluun kehitetty sovellus, joka tarjoaa toimintoja 3D-suunnitteluun, simulointiin ja dokumentointiin. Sen ensimmäinen versio julkaistiin vuonna 1999. (Trevor English. 2017. Autodesk Inventor Blog)



Kuva 6. Autodesk Inventor kuvituskuva. (<https://www.sapectr.com>)

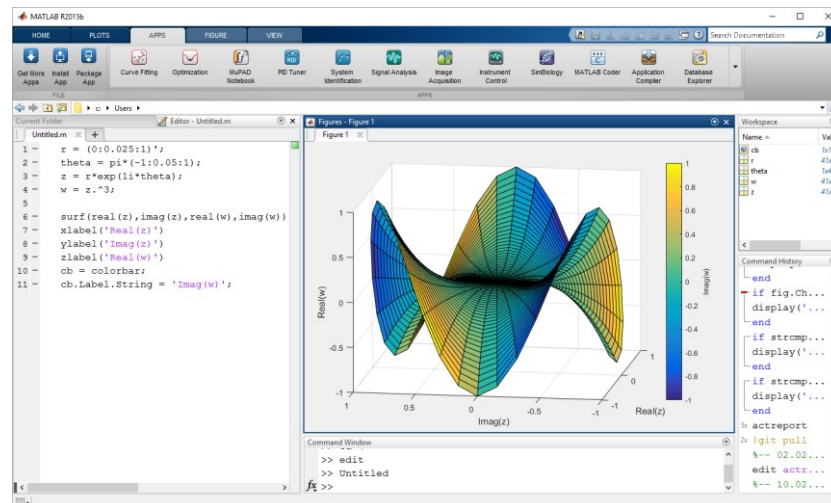
Siemens NX on Siemens PLM Softwaren kehittämä suunnitteluun, tuotantoon ja analysointiin sekä simulointiin kehitetty sovellus. NX tarjoaa useita liitännäismahdollisuuksia muiden sovelluksien välille lisäten sen käyttöulottuvuuksia

oman ekosysteeminsä ulkopuolelle. NX on suora kilpailija Autodesk Inventorin sovelluksen kanssa. NX julkaistiin sen nykyisessä muodossaan ensimmäisen kerran vuonna 2005 ja on saanut uuden julkaistavan version melkein vuosittain. (Siemens NX product website)



Kuva 7. Siemens NX12 kuvituskuva. (<https://blogs.sw.siemens.com>)

Matlab on The Mathworks -yhtiön numeeriseen laskentaan tarkoitettu sovellus. Sen kehitys alkoi 1970-luvulla ja se kaupallistettiin vuonna 1984. Se on laajalti eri yritysten ja instituutioiden sekä korkeakoulujen käytössä. Matlab yksi tärkeimmistä ominaisuuksista on sen graafinen ympäristö, Simulink. Simulinkin avulla voidaan mallintaa ja simuloida dynaamisia malleja useilla eri tavoilla. Simscape on Simulinkin lisäosa, jonka avulla voidaan toteuttaa 3D-mallinnusta myös Matlabin sisällä. Simscape-lisäosa tarjoaa mekaanisia, sähköisiä ja pneumaattisia sekä hydraulisia mallinnusmahdollisuuksia. (About mathworks verkkosivu)



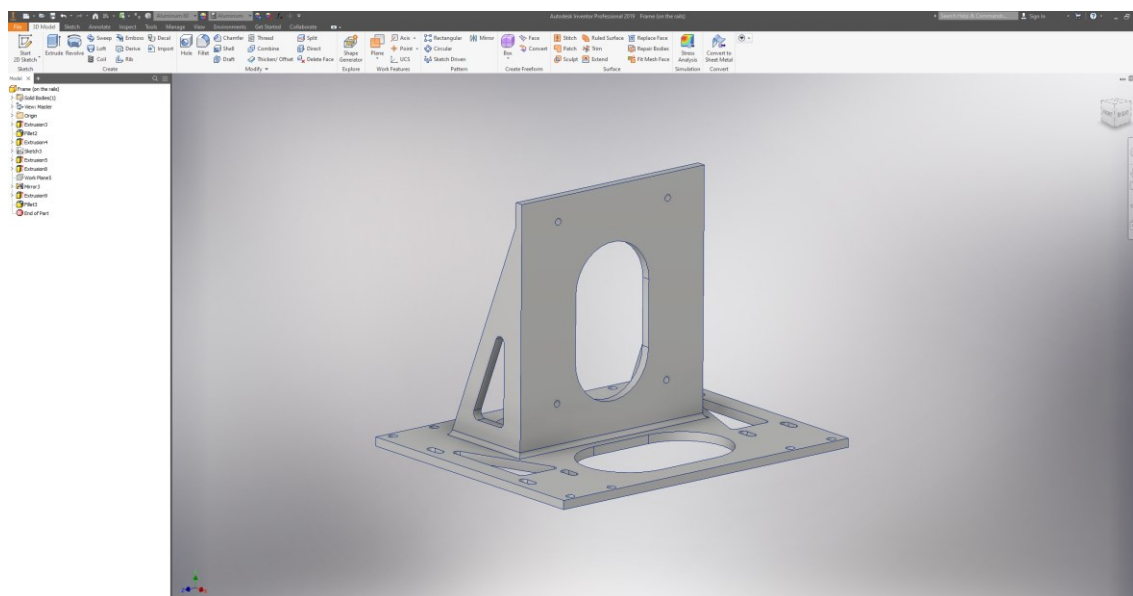
Kuva 8. Mathwork Matlab kuvituskuva. (<https://upload.wikimedia.org>)

Tässä työssä käytettiin seuraavia ohjelmia, niiden versioita sekä kirjastoja:

- Autodesk Inventor Professional 2019
 - o +Simscape Multibody Link -lisäosa
- Siemens NX 12
- Matlab R2019b
 - o Simulink versio 10.0
 - o Simulink Control Design 5.4
 - o Simscape versio 4.7
 - o Simscape Multibody versio 7.0
 - o Simscape Driveline versio 3.0
 - o Simscape Electrical versio 7.2
 - o +Simscape Multibody Link -lisäosa

3 DIGITAALISEN KAKSOSEN LUOMINEN

Digitaalisen kaksosen luominen aloitettiin piirtämällä kaikki laitteen osat Autodesk Inventor -sovelluksella, josta ne tuotiin yleisessä step-muodossa Siemens NX puolelle.

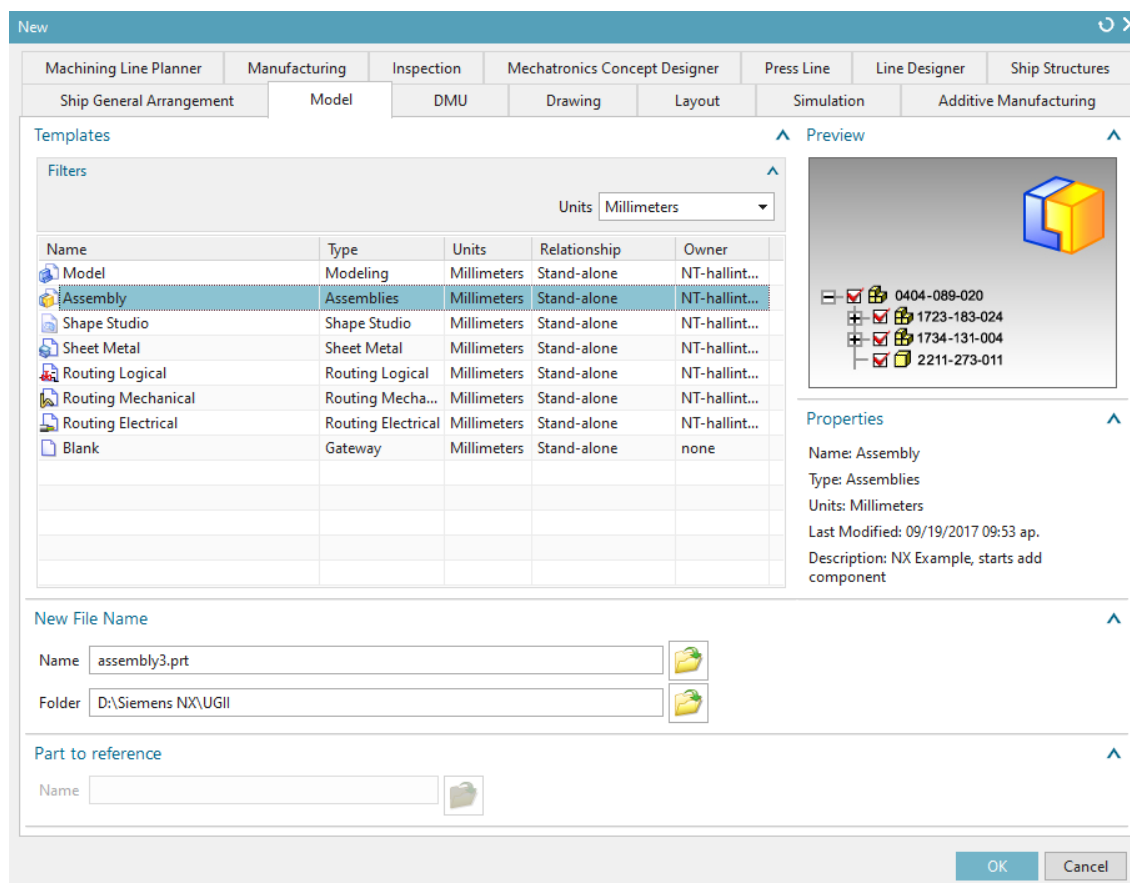


Kuva 9. Laitteen runko Inventor-ympäristössä.

Siemensin puolella kappaleista luotiin kokoonpano, missä hienosäädettiin ja viimeisteltiin kokoonpanon toiminnallisuus. Kokoonpanon jälkeen malli vietiin Siemens NX Simcenter motion -puolelle, missä määriteltiin simuloinnin kannalta oleelliset kappaleiden vuorovaikutussuhteet, ympäristö ja luotiin yhteys Matlab Simulink -puolelle. Matlab toimii tässä työssä isäntäohjelmana, jossa simuloinnin laskennallinen työ suoritettiin. Siemens NX toimii käytännössä simuloinnin näyttöpäätteenä ja joidenkin toiminnallisuuksien kuten törmäysten ja kappaleiden massan lähteenä Matlabia varten. Simulointi tapahtuu näiden kahden ohjelman välisenä yhteistyönä.

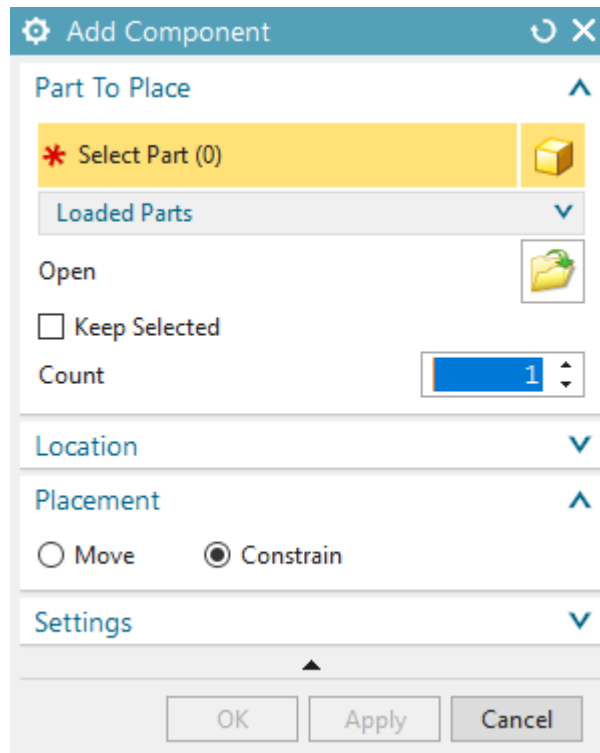
3.1 Siemens NX assembly

Autodesk Inventorilla piirretyt kappaleet tuodaan Siemensin käytettäväksi avaamalla kappaleet ja tallentamalla ne uudelleen siemens part -tiedostoiksi. Tämän jälkeen kappaleet on tuotavissa assembly-ympäristöön



Kuva 10. Siemens NX toimintojen yleisnäkymä uutta projektia aloittaessa.

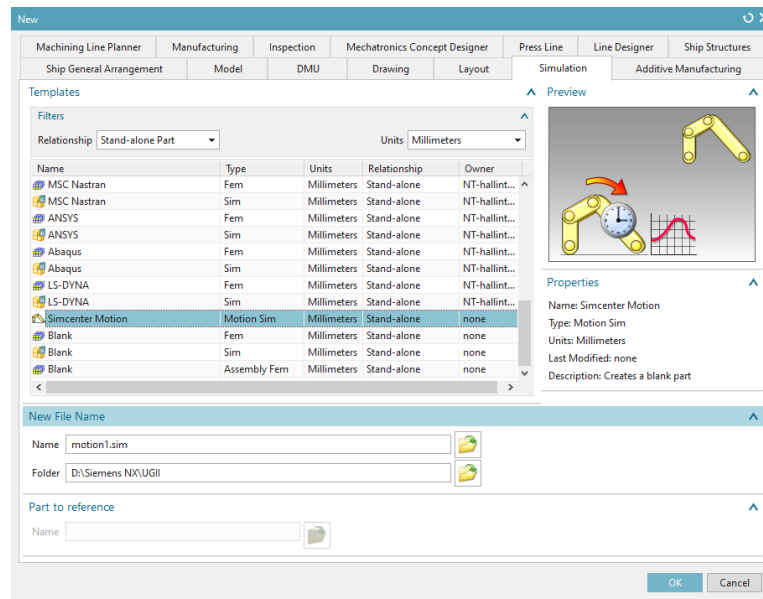
Assembly-ympäristö on tarkoitettu kokoonpanojen rakentamiseen. Täällä kappaleet tuodaan yksitellen tai yhdessä add component-toiminnolla. Kappaleen tuonnin jälkeen sille asetetaan sen paikkaa sitovat rajoitukset. On hyvä ottaa huomioon kappaleen liikkeen kannalta oleelliset liikkeet ja niiden luonteet. Liikkuvia kappaleita kiinnittäessä niiden liikeratojen tulisi olla liikkeensä suuntaan vapaita eikä näin ollen niitä tule kiinnittää ylimääräisillä rajoituksilla toisiinsa. Kappaleet hakevat niiden paikkansa kiinnittäessä niitä toisiinsa automaattisesti. Kappaleiden liikuttaminen on myös mahdollista erillisellä move component-työkalulla.



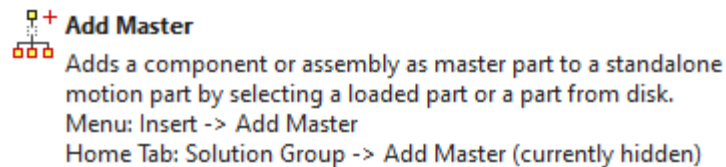
Kuva 11. Siemens NX komponenttien lisäämiseen käytetty työkalu.

3.2 Simcenter Motion

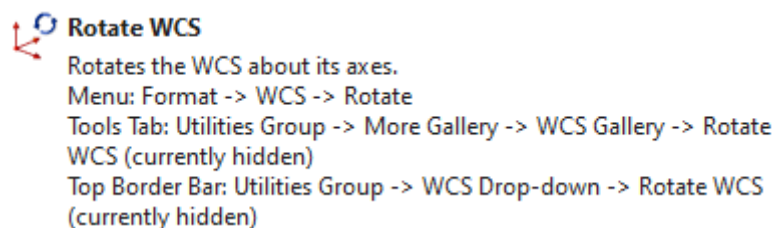
Aikaisemmin rakennetun kokoonpanon liikkeitä voidaan simuloida "Motion" -ympäristössä. Täällä kokoonpanoon voidaan määritellä muun muassa liikeratoja, painovoima sekä liike- ja lepokitka materiaalien välille. Tyhjään motion-malliin kokoonpano voidaan tuoda add master-komennolla. Kokoonpanoa voidaan kiertää ja kääntää tarvittaessa haluttuun asentoon (kuva 14).



Kuva 12. Siemens NX toiminnallisuusikkuna uutta projektia luotaessa.



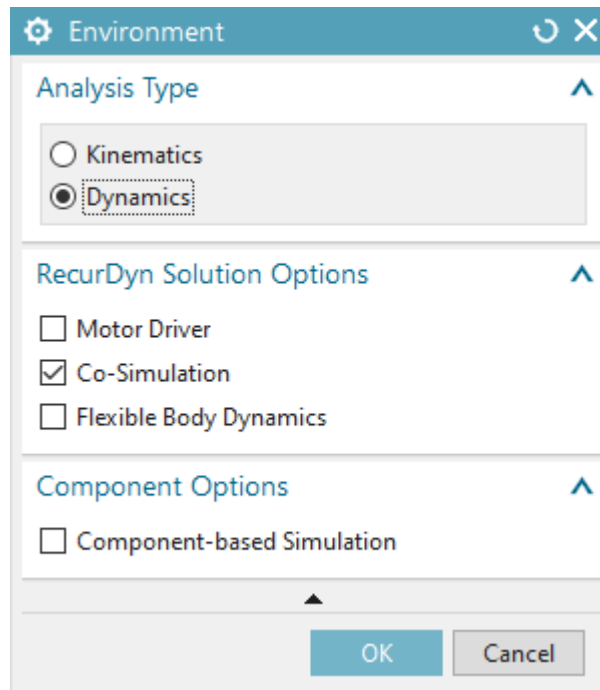
Kuva 13. Kokoonpanon tuomiseen käytetty työkalu.



Kuva 14. Kokoonpanon koordinaattien kääntöön käytetty työkalu.

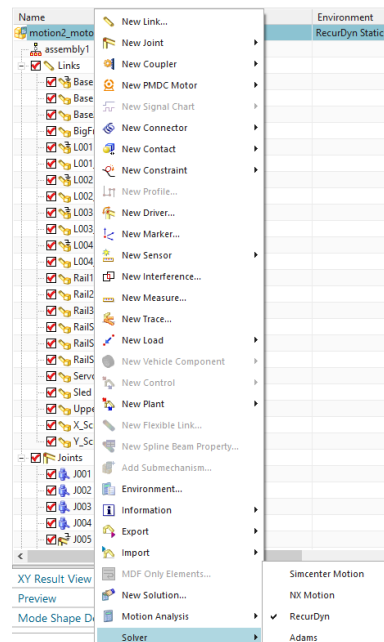
3.2.1 Simulointiympäristö

Simulointiympäristössä kokoonpanolle valitaan dynaaminen analyysityyppi environment-asetuksista. Siemens NX:n ja Matlab Simulinkin yhteissimulaatioissa käytetään RecurDyn-laskentamallia ja jotta yhteissimulaatio voidaan suorittaa tulee samasta asetussivusta laittaa co-simulation päälle.



Kuva 15. Siemens NX simulaation ympäristön määrittämisen työkalu.

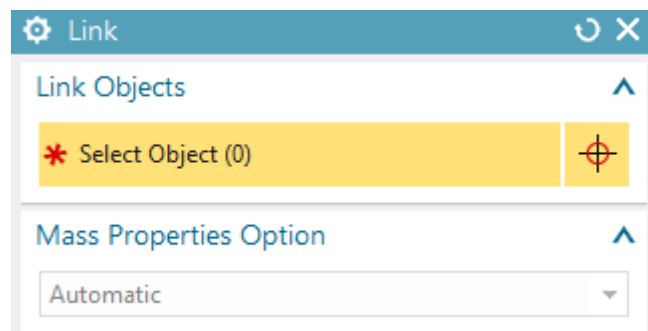
Kun ympäristö on määritelty, voidaan itse kokoonpanon takaa käydä vaihtamassa mallille RecurDyn-ratkaisija päälle (kuva 16). Jotkin Siemensin Motion-ympäristön toiminnoista eivät ole käytettävissä yhteensopimattomuutensa vuoksi tätä ratkaisijaa käytettäessä.



Kuva 16. Siemens NX simulaatiomallin ratkaisijan valinta.

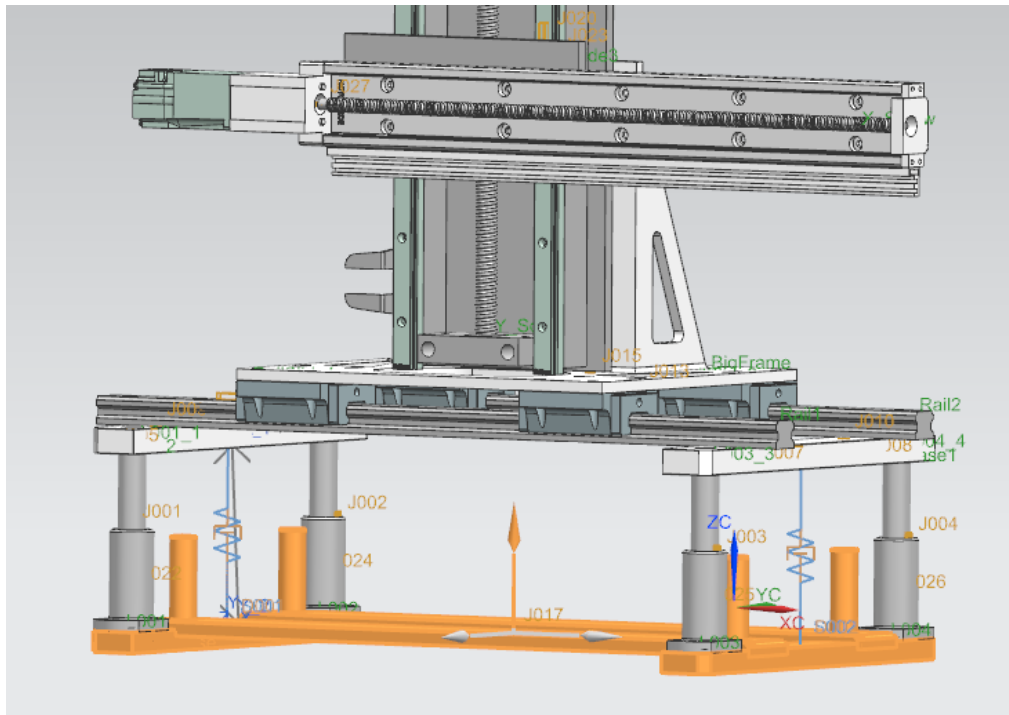
3.2.2 Linkit

Motion mallin rakentaminen aloitetaan linkkien määrittämisellä. Linkeillä on tarkoitus määrittää ne kokoonpanon osat, jotka ovat vuorovaikutuksissa keskenään. Tässä tapauksessa ensimmäinen linkki on pohjarunko. Pohjarungon linkki koostuu useista kappaleista, sillä näillä ei ole vuorovaikutusta keskenään, eikä näin ollen ole oleellista luoda näistä eri linkkejä.



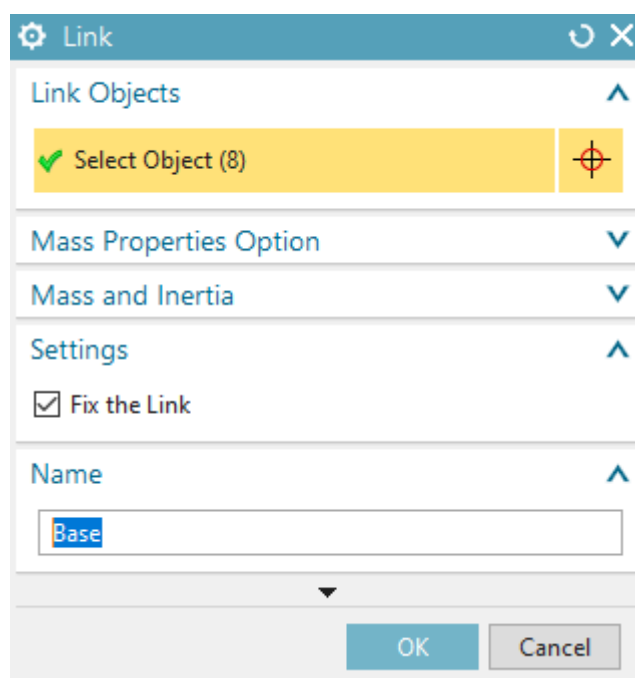
Kuva 17. Linkkien määrittäminen.

Samalla periaatteella keskeisimmät, suuret, toisistaan liikkumattomat rungon osat on linkitetty keskenään yhteen. Toisissaan vuorovaikutuksissa olevat osat on luotu omiksi linkeikseen. Jos kokoonpanon jokaisen komponentin luo omiksi linkeikseen, voi se aiheuttaa turhan paljon laskelmointityötä simuloinnin aikana. Myös materiaaliasetuksista riippuen liitokset saattavat joustaa tarpeettomasti.



Kuva 18. Siemens NX simulaatiomallin yleisnäkymä. Base-linkki.

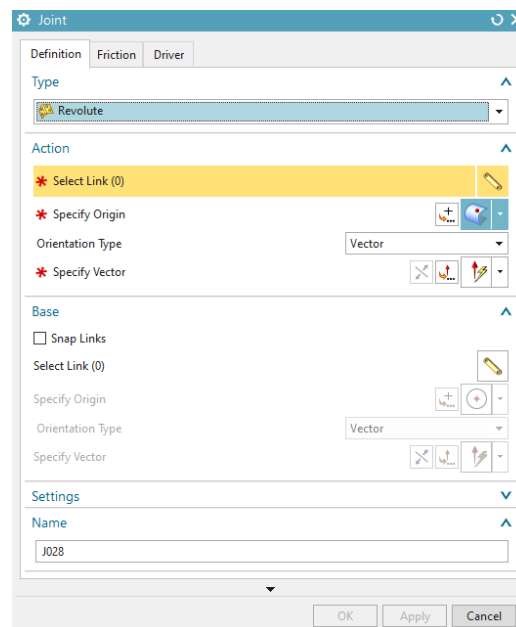
Simuloinnin aikana malliin vaikuttaa muun muassa painovoima. Painovoima määritellään myöhemmin, mutta mallille on oleellista kiinnittää jo tässä kohtaa joitain sen osia paikalleen, jottei se valu ulos ympäristöstä. Linkkien luonnin aikana voidaan kappale kiinnittää paikoilleen fix the link -toiminnolla. Tässä tapauksessa ensimmäinen linkki "Base" on määritetty kiinnitetyksi linkiksi.



Kuva 19. Base-linkin määritysikkuna.

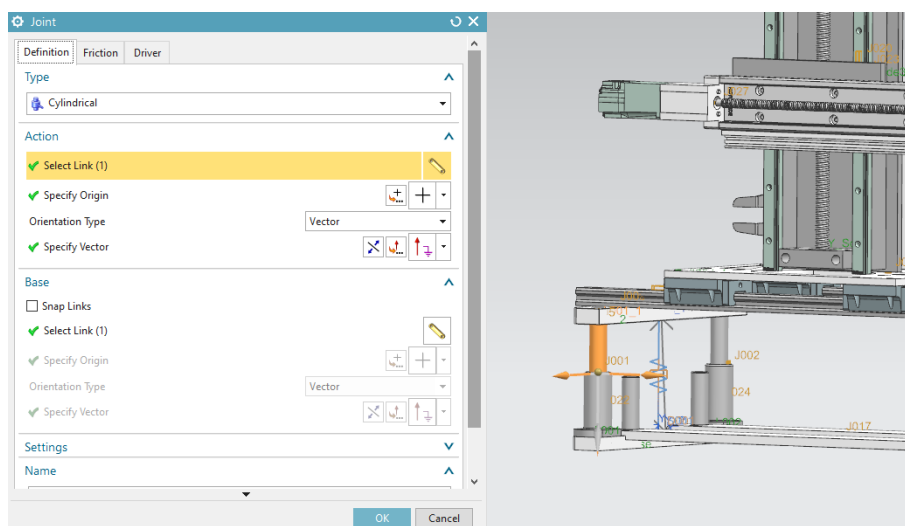
3.2.3 Liikeradat

Kun linkit on määritelty, on motion-mallille määriteltävä linkkien väliset vuorovaikutussuhteet. Joint-toiminnolla linkkien välille voidaan luoda liikkeitä, kitkaa sekä tarvittaessa liikuttava voima. Driver-välilehden ominaisuuksia ei käytetty, sillä liikuttavat voimat tarjotaan mallille Matlabin avulla.



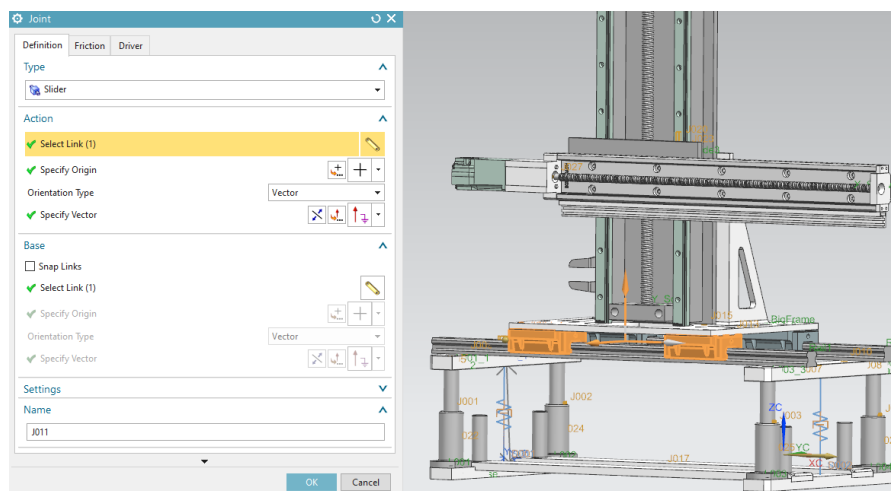
Kuva 20. Vuorovaikutussuhteiden määrittämisen työkalu.

Jokainen pohjasylinteri on vuorovaikutuksessa pohjarungon, sylinterin sisällä olevan akselin ja lineaarikiskon alla olevan rungon kanssa. Akseli on sylinterimäisessä vuorovaikutuksessa sylinterin kanssa, sekä kiinteästi kiinni ylemmässä rungossa.



Kuva 21. Esimerkki vuorovaikutussuhteesta.

Ruuvijohtimet ovat saranaliitoksella kiinni rungossa. Ne ovat ruuviliitoksella vuorovaikutuksessa myös liikuteltavan kelkan kanssa ja tähän on määriteltä myös kierteen nousu. Kierteen nousu on myös piirrettyä kappaleeseen, mutta tällä ei vuorovaikutussuhteiden kannalta ole merkitystä kuin visuaalisesti.



Kuva 22. Esimerkki vuorovaikutussuhteesta.

3.2.4 Liikeratojen rajoittaminen

Kokoonpanon eri liikeratoja voidaan rajoittaa törmäyskontaktien avulla. 3D contact -toiminnolla voidaan määrittää törmäysvuorovaikutus eri kappaleiden välille. Määrittelemällä törmäyspinnat esimerkiksi Y-akselin kelkan ja rungon välille voi-

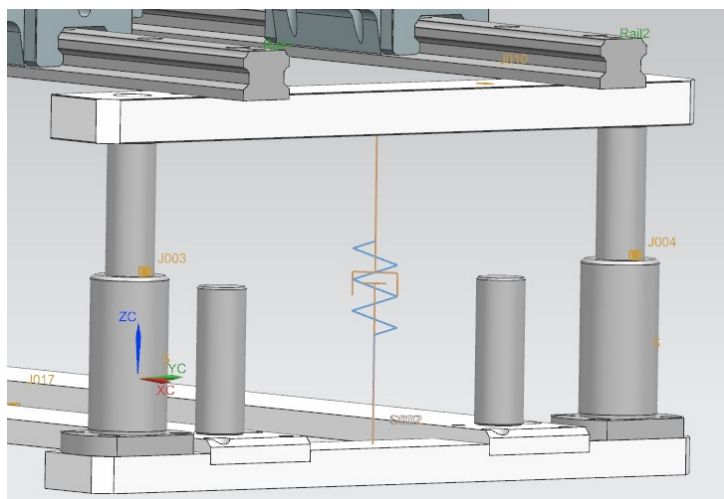
daan estää kelkan kulkeutuminen ulos sen todelliselta liikealueelta. Tätä toimintoa käyttäessä huomattiin simulointiaikojen merkittävä kasvu sellaisissa tilanteissa, joissa törmäys tapahtuu. Tämän johdosta toiminto otettiin pois käytöstä simulointien ajaksi. Törmäyskontaktit ovat mielekästä ottaa käyttöön, kun törmäys on simuloinnin kannalta oleellinen, kuten tuloksia viimeisteltäessä.

Taulukko 1. Simulointiajat 3D-kontaktin kanssa.

3D kontakti päällä	03.45 min
3D kontakti pois päältä	00.17 min

3.2.5 Jouset

Laitteen toiminnan yksi pääpiirteistä on sen jousiripustein rakenne. Laitteen runko lepää jousten päällä, jolloin sen on mahdollista joustaa nopean liikkeen tai ulkoisen voiman seurauksena. X-akselin suuntainen liike on myös jousiripustein. Tällöin kaikki laitteen suunnat antavat myöden tarpeeksi suuren voiman kohdistuessa siihen.



Kuva 23. Kuva kokoonpanon jousivaimentimesta.

Laitteen jouset voidaan määrittää omalla jousityökalulla. Työkalun avulla jousi kiinnitetään kahden linkin välille sekä määritellään jousivakio. Mallinnettavan laitteen jousen jousivakio mitattiin epävirallisissa olosuhteissa vaa'an avulla. Jousen käyttäytymistä kuvattiin tässä vaiheessa jousivakion arvolla, mutta sen

käyttäytymistä voidaan kuvata myös funktion muodossa jousen tarkempaa mallintamista varten.

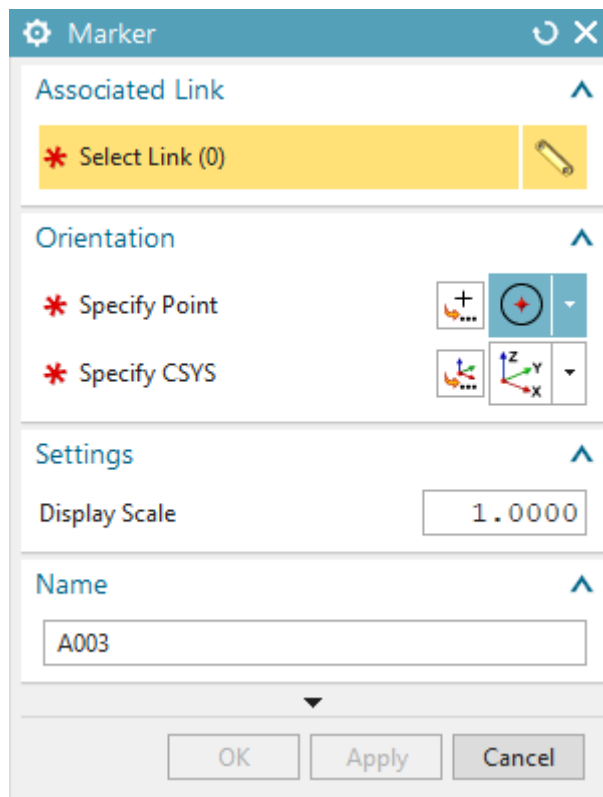
Kuva 24. Jousi-vaimentimen määrittämisen työkalu.

Jouselle voidaan määrittää myös sen automaattisesti määritetystä pituudesta poiketen jokin muu pituus, joka luonnollisesti vaikuttaa jousen käyttäytymiseen simuloinnin aikana. Tämä jätettiin tällä kertaa omaan automaattiseen arvoonsa. Jousen rinnalle on hyvä luoda samalla vaimennin, sillä erinäiset kitkavoimat vaimentavat jousen värähtelyä, vaikka erillistä vaimenninta laitteessa ei olekaan.

3.2.6 Sensorit

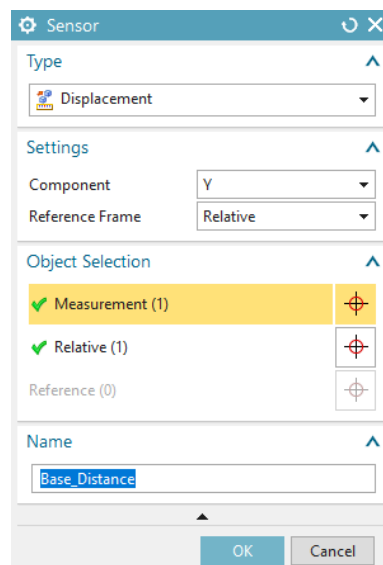
Sensorit toimivat myöhemmin rakennettavassa simulink-mallissa simuloitavan mallin ulostuloina. Sensoreiden tarkoituksena on mitata kappaleen fysikaalisia ominaisuuksia, kuten nopeutta, paikkaa tai kiihtyvyyttä. Sensoreita varten on

luotu myös koordinaattipisteitä. Koordinaattipisteiden avulla sensoreiden mittapisteen on helppo määrittää ja pisteiden paikka on muutettavissa tarvittaessa myöhemmin.



Kuva 25. Koordinaattipisteiden työkalu.

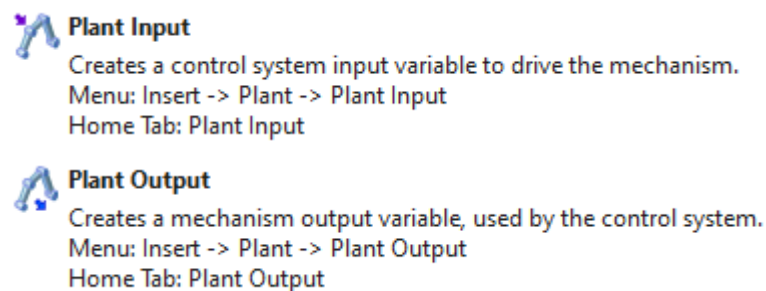
Tässä mallissa on käytetty kuutta sensoria mittaamaan x- ja y-akselien etäisyyttä, rungon etäisyyksiä sekä molempien ruuvijohtimien kiertokulmaa.



Kuva 26. Sensorien työkalu.

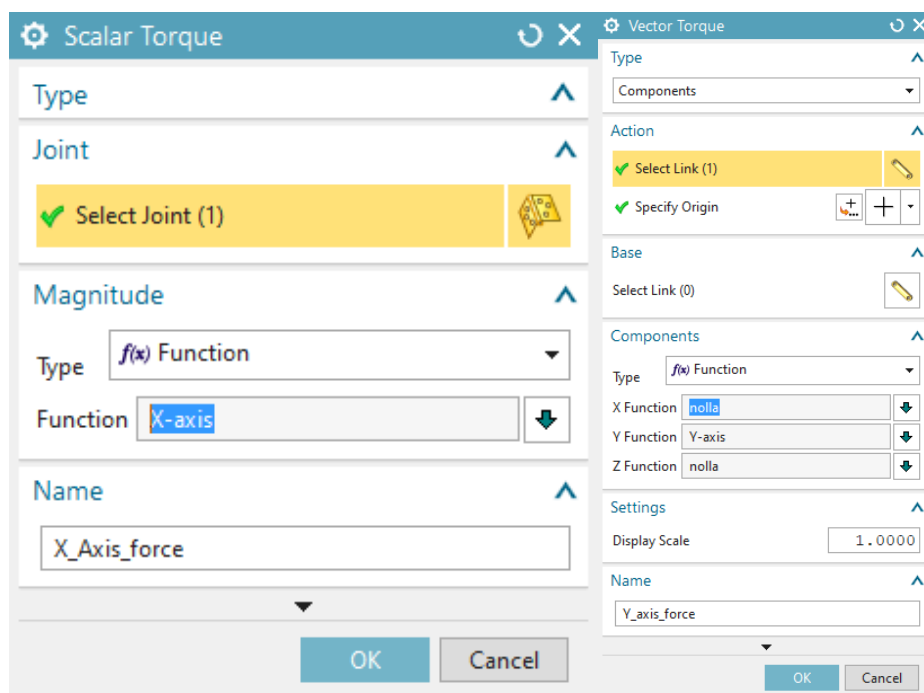
3.2.7 I/O

Mallin sisään- ja ulostulot määritellään käyttäen plant input ja plant output -toimintoja. Tähän malliin luotiin kolme sisääntuloa; molemmille moottoreille sekä ulkoiselle runkoon kohdistuvalle voimalle, jota käytettiin testimielessä moottorin toiminnan tarkkailuun ja säätämiseen.



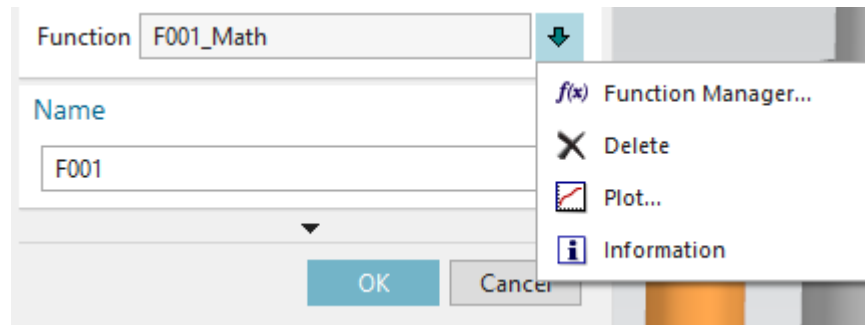
Kuva 27. RecurDyn-ratkaisijan kanssa käytetyt I/O-työkalut

Malliin kohdistuvat voimat määritellään joko skalaari- tai vektorivoimiksi. Moottoreiden voimat voidaan määritellä kummalla tahansa. Vektorivääntömomenttia käyttäessä suunnat, joilla ei ole momenttia merkitään nollaksi. Nollaksi voima merkitään funktioeditorissa luodun, nolla-arvoisen, funktion avulla.

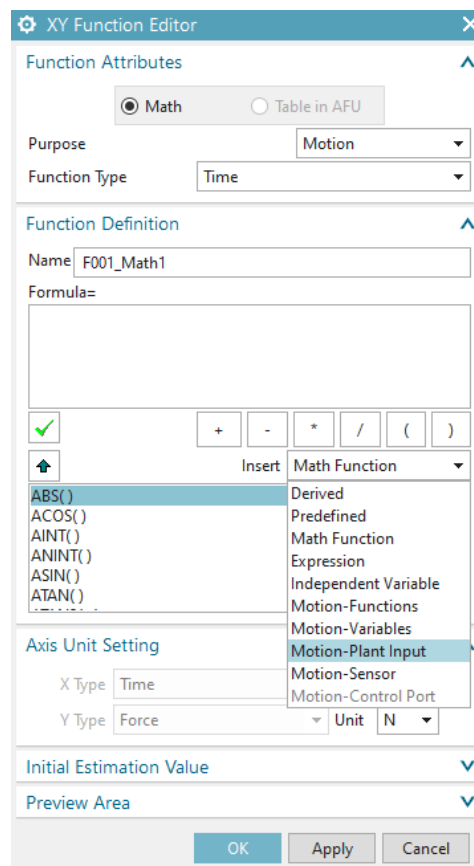


Kuva 28. Skalaari- ja vektorivoimatyökalut.

Voimat liitetään niihin kuuluviin sisääntuloihin funktioiden avulla. Funktioeditorin sisällä luodaan uusi funktio ja valitaan insertiksi ”Motion-Plant Input”. Editori listaa valittavissa olevista inputfunktioista, joita raahaamalla kaavakenttään saadaan linkki luotua.



Kuva 29. Funktioeditorin käyttö.



Kuva 30. Funktioeditorin näkymä.

Funktioeditorissa sisääntulojen kanssa voidaan käyttää myös matemaattisia kaavoja. Tässä mallissa haluttiin kuitenkin vain linkittää voimat omiin sisääntu-

loihinsa. Funktioita määrittäessä voimien yksiköihin tulee kiinnittää huomiota. Simulinkin puolella on tarkoitus tuottaa vääntömomentin arvoja, joiden yksikkö on N/m. Siemens NX oletuksena määrittää vääntömomentin yksiköksi N/mm.

3.2.8 Ratkaisumalli

Viimeiseksi mallille luodaan ratkaisumalli, jossa simulointi suoritetaan. Ympäristön luominen tapahtuu solution -toiminnolla. Ympäristössä valitaan analyysityyppi Control/Dynamics. Simulointiajaksi on valittu 10 sekuntia ja askelten määrä on 500. Tämä on mielivaltaisesti valittavissa. On kuitenkin hyvä huomata, että simulointiajalla ja askelten määrällä on suora yhteys simuloinnin suorittamiseen kuluvaan aikaan. Pidemmällä simulointiajalla saadaan pidempi otanta laitteen toiminnasta ja askelmäärällä voidaan vaikuttaa simuloinnin tarkkuuteen.

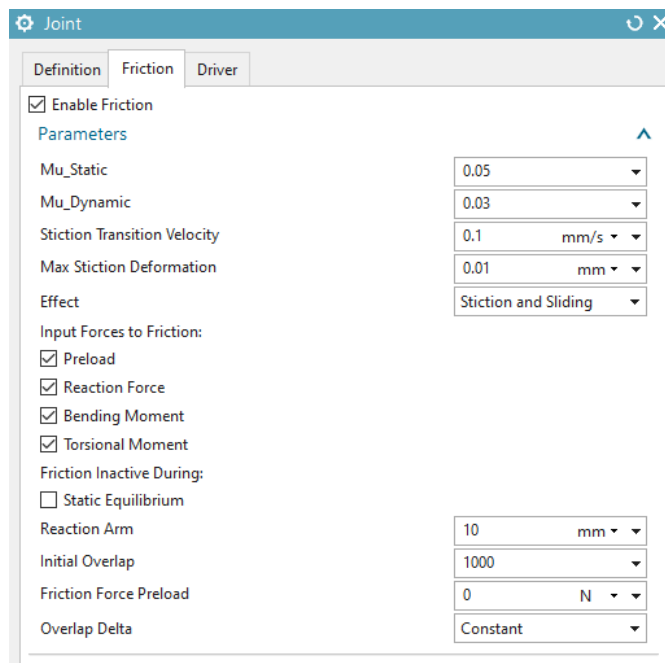
Painovoima määriteltiin kokoonpanomallin koordinaattiakselin Z suuntaiseksi. Jos koordinaattiakseli on mallissa toisin päin, tai muusta syystä halutaan painovoiman vaikuttavan toisin, voidaan se tässä kohtaa määrittää. Co-simulation -kohdassa näyteajaksi määritellään 0.001 sekuntia. Samaa näyteaikaa on hyvä pitää myös Simulinkin puolella johdonmukaisten tulosten saavuttamiseksi. Muut ympäristön asetukset jätettiin oletusarvoihinsa.

Kun ympäristö on määritelty, luodaan Simulink-tiedosto valitsemalla Solve. Isäntäsovellukseksi valitaan Simulink, sillä tässä työssä on tarkoitus suorittaa laskennallinen työ Matlabin puolella.

3.2.9 Kitka

Liikeratojen määrittelyssä on mahdollista asettaa liikkeiden välille kitkavoimia. Tässä mallissa kitkavoimat arvioitiin ilman sen tarkempia mittauksia. Kitkakertoimien arvoiksi määriteltiin sellaiset arvot, joilla malli on tasapainossa, kun siihen

ei kohdistu ulkoisia voimia. Kitkavoimien määrittäminen arvioperusteisesti oli simuloinnin kannalta oleellista, sillä muuten malli on herkkä oskilloinnille. Ilman kitkan määrittystä, mallista on miltei mahdoton saada stabiili kokonaisuus.



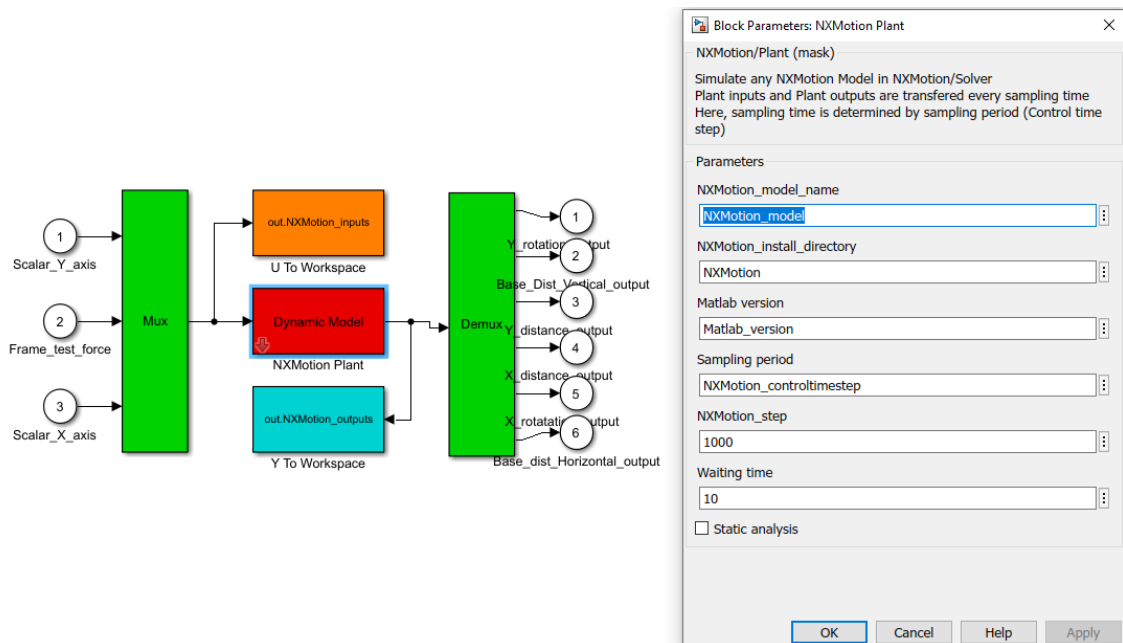
Kuva 31. Kitkavoimien määrittäminen kappaleiden välille.

4 SIMULINK

Siemens NX puolella luotiin Simulink-malli, joka tallentuu samaan kansioon muiden mallin tiedostojen kanssa. Malli sisältää kaksi tiedostoa: Solution_1_Plant.m sekä Solution_1_PlantIO.m.

Plant.m tiedoston sisällä kommentoidaan pois viimeisin rivi. Tiedosto tallennetaan ja suoritetaan. Matlab avaa ikkunan, jonka sisällä on nyt edellisessä kappaleessa määritelty malli. Mallin vasemmalla puolella on sisääntulot sekä oikealla puolella ulostulot.

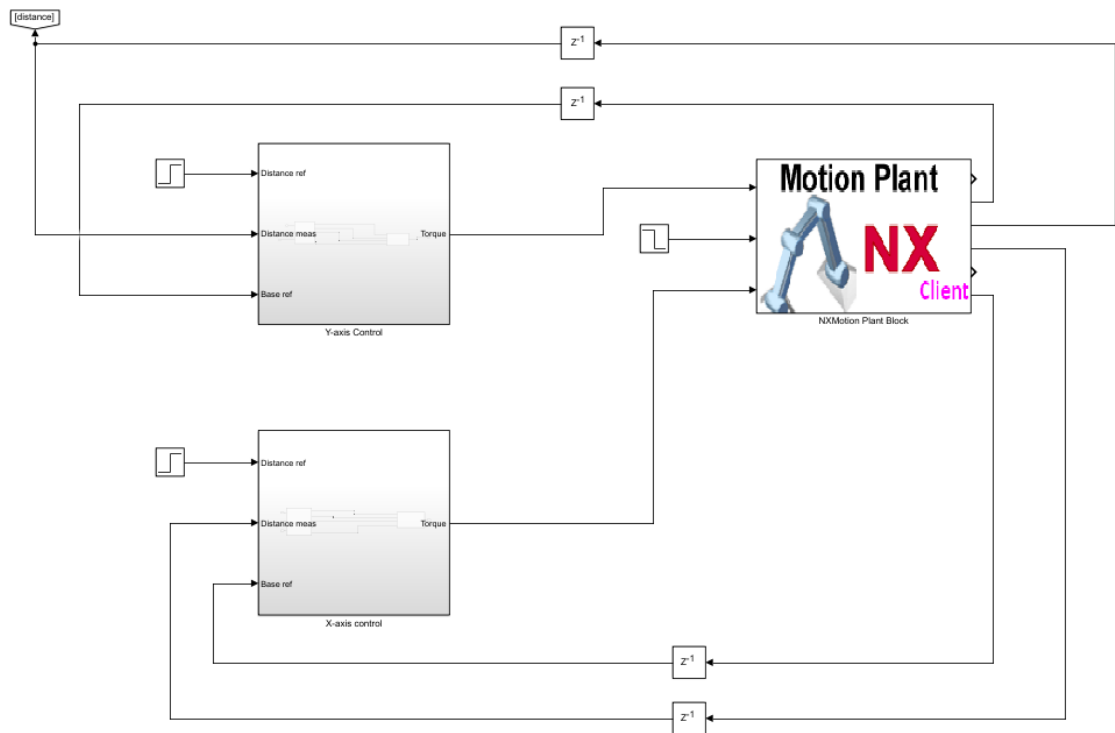
Siemensin luomat tiedostot jätetään sellaisikseen. Sitten luodaan uusi tyhjä malli, jonne Siemensin Motion Plant siirretään. Alkuperäinen tiedosto tallennetaan ja suljetaan. Simulink-mallia ei rakenneta Plant.m-tiedoston sisälle. Mallin sisältä on määriteltävä dynaamisen mallin askelmäärät ja odotusaika (kuva 32). Täällä askelmäärä määrittää yhteissimulaation ratkaisujen määrän eikä ole korrelaatiossa motion-ympäristössä määritellyn askelmäärän kanssa. Odotusaika on aika sekunteina, kuinka kauan yhteyden uudelleenmuodostumista odotetaan, jos se katkeaa NX ja Matlabin välillä. Tässä mallissa on käytetty arvoja 1000 askelta ja 10 sekuntia.



Kuva 32. Motion Plant-asetukset.

Jos Siemensin puolella malliin tehdään muutoksia, tulee solve suorittaa uudelleen, jolloin toistetaan äskettäin suoritettut vaiheet ja korvataan vanha Motion Plant uudella.

Viiveet (Z^{-1}) kuvastavat fyysisen laitteen antureiden mittausviivettä (kuva 33), sekä poistavat *algebraic loop*-virheiden ilmaantumisen laskennassa.



Kuva 33. Simulink-malli.

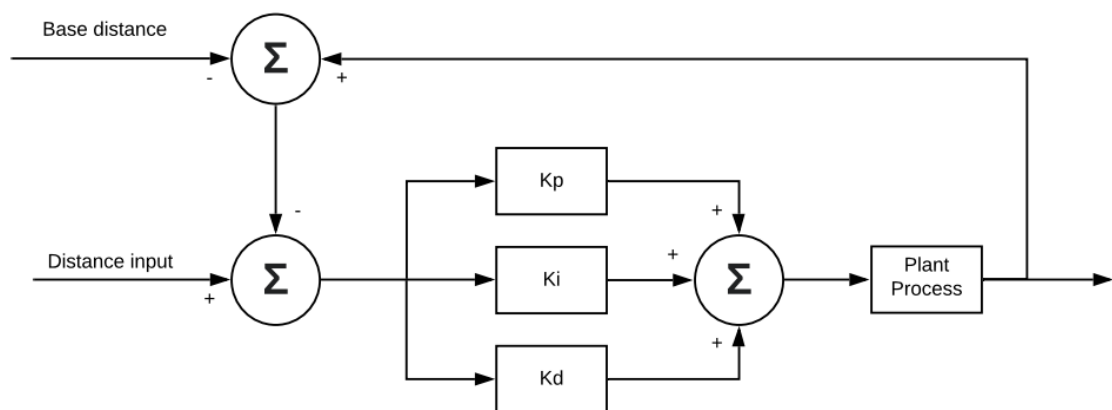
4.1 PID säätö

Moottorin vääntömomenttia ohjataan PID-säätimen avulla. PD-säädin saa säädettävän arvonsa etäisyyden ja ohjearvonsa erotusfunktioista (kuva 35). Säätimen tehtävänä on saattaa tämä erotus nollaan mahdollisimman tehokkaasti. Säädin on rajoitettu antamaan arvoja 0 ja 10 välille. Matlabissa valmis PID-säätinpohja mahdollistaa monia integroituja ominaisuuksia säätimen toimintaan, kuten ulostulon saturaation määrittämisen.

Säätöä tehdessä testattiin säätää PID-säätimen arvoja lookup-taulukon avulla. Tulokset olivat lupaavia mutta eivät johdonmukaisia, joten niitä ei esitetä tässä työssä.

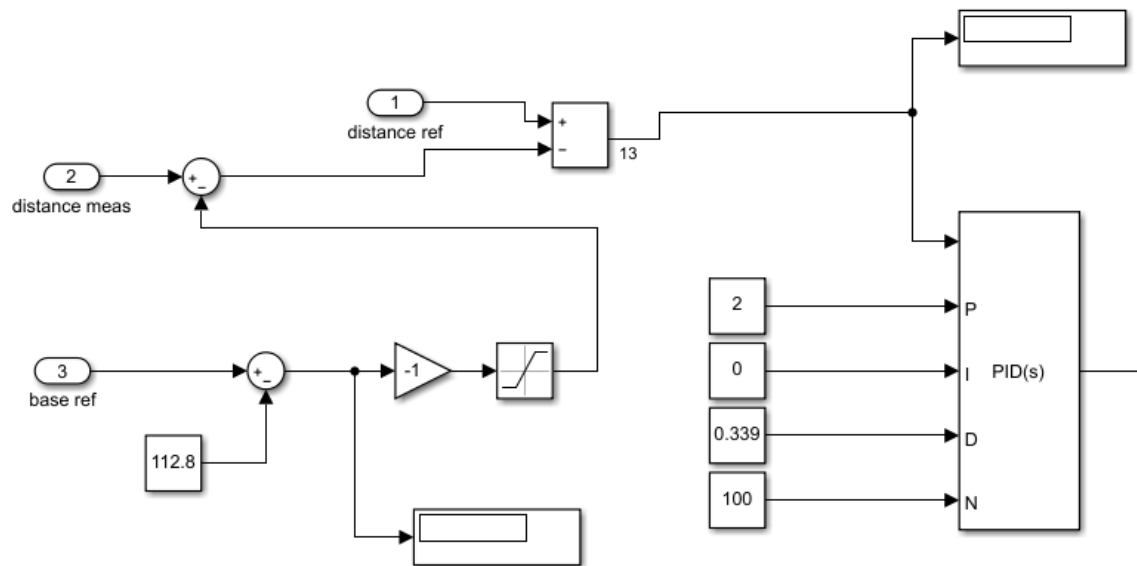
Tässä työssä havaittiin PID-säädön kankeus laitteen toiminnassa. Havaittiin, että kelkkaa liikuteltaessa sen kulkema matka vaikutti sen tarkkuuteen ja toimintaan. Myös painovoiman vaikutus oli havaittavissa. Näissä tilanteissa aikaisemmin mainittu adaptiivinen säätö voi osoittautua lupaavaksi tarkkuutta parantavaksi tekijäksi.

4.2 Värähtelyn huomioonottaminen PID-säädössä



Kuva 34. Moottorin PID-säädön sisempi sekä ulompi säätökaavio.

Laitteen PID-säädön yhtenä ominaisuutena oli ottaa huomioon sen jousiripustetun rakenteen värähtely moottorin paikoituksessa. Laitteen rungon värähtelyn etäisyys otettiin yksinkertaisella menetelmällä erotusfunktion avulla. Kuvassa 34 on kuvattu moottorin säädön sisempää sekä ulompaa säätöä. Kaaviossa ei ole esitetty anturin mittaaman etäisyysarvon nollaamista. Anturin antamat arvot rajoitettiin ± 10 millimetrin välille, sillä rajoittamattomana se tuotti tarpeettoman rajuja muutoksia PID-säätimen toimintaan.

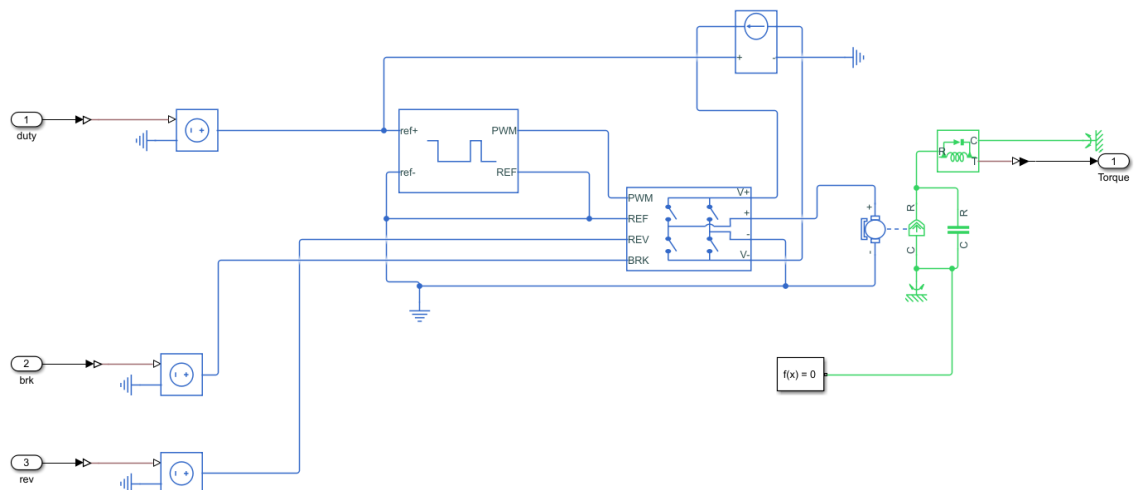


Kuva 35. Värähtelyn huomioonottaminen PID-säädössä Simulink-mallissa.

Säätöä suunnitellessa on otettava huomioon antureiden aiheuttama viive. Pahimmillaan viive viivästyttää liikkeen kompensointia liikaa, jolloin korjaavat liikkeet lisäävät värähtelyä. Tämän mallin viiveet ovat 1 millisekuntia, joka ei kuvaa todellisen anturin tosiasiallista vasteaika.

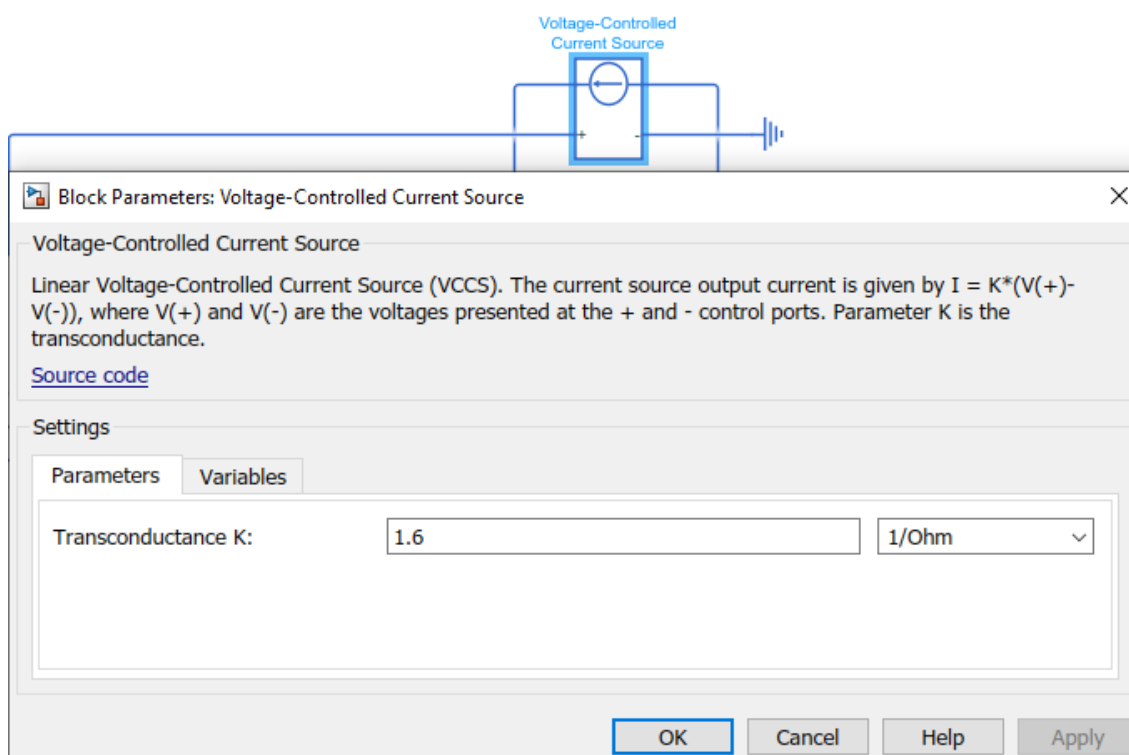
4.3 Moottorin mallinnus

Moottorin mallinnukseen Simulink tarjoaa valmiita kirjastoja, joiden avulla on tarkoitus pystyä mallintamaan yleisimmät moottorityypit. Tässä työssä laitteeseen kiinnitetyt AC servomoottorit mallinnettiin PWM-ohjattujen DC-moottoreiden avulla yksinkertaisemman mallinsa vuoksi.



Kuva 36. Moottorin jännitelähteet, moottori sekä H-silta.

Useita eri moottorityyppejä testattiin silmäilemällä niiden käyttäytymistä ja todettiin DC-moottorimallin olevan riittävä mallinnustarkkuus. Oleellista mallin luonnissa oli moottorin tuottaman väännön luonne eikä niinkään sen rakenne. Moottorin mallinnuksen perustana käytettiin laitteessa kiinni olevan servomoottorin vääntömomenttikäyrää ja sen tehoarvoja vastaavaa DC-moottorin datalehden arvoja. Moottorin maksimivääntömomentti halutaan saavuttaa hetkellisesti, kuten liikkeellelähdössä, jonka jälkeen moottorin tulisi palata tuottamaan nominaalista vääntömomenttiaan. Tämä saavutetaan hyvin tuloksin suhteuttamalla moottorille saatavissa oleva virta PID-säätimeltä tulevaan säätöarvoon sopivalla transkonduktanssin arvolla (kuva 37).

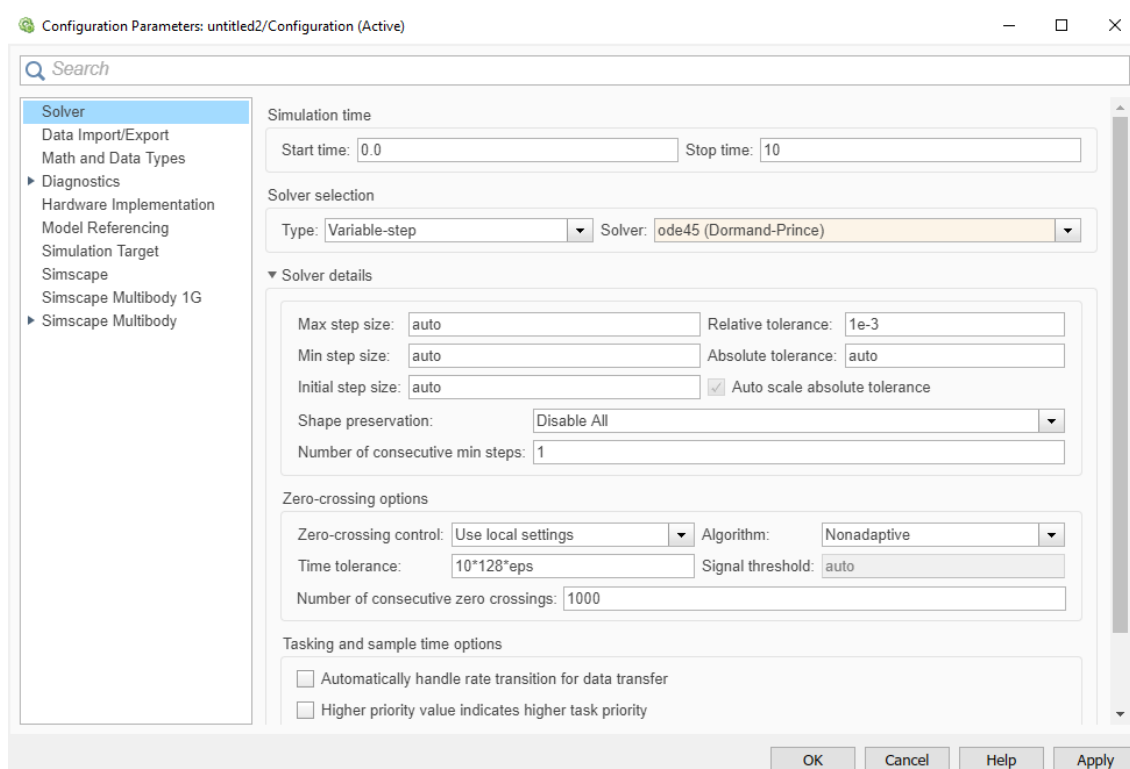


Kuva 37. Jännitesäädettävä virtalähde moottorille.

4.4 Simulointi

Simulointi suoritetaan suorittamalla ensin simulaatio Matlabin puolella, jonka jälkeen tulokset on katsottavissa Siemensin puolella results-välilehdellä. Simulointiin kuluva aika riippuu simuloitavan mallin monimutkaisuudesta, tarkkuudesta, käytettävästä ratkaisijasta ja yksittäisten komponenttien simulointiasetuksista.

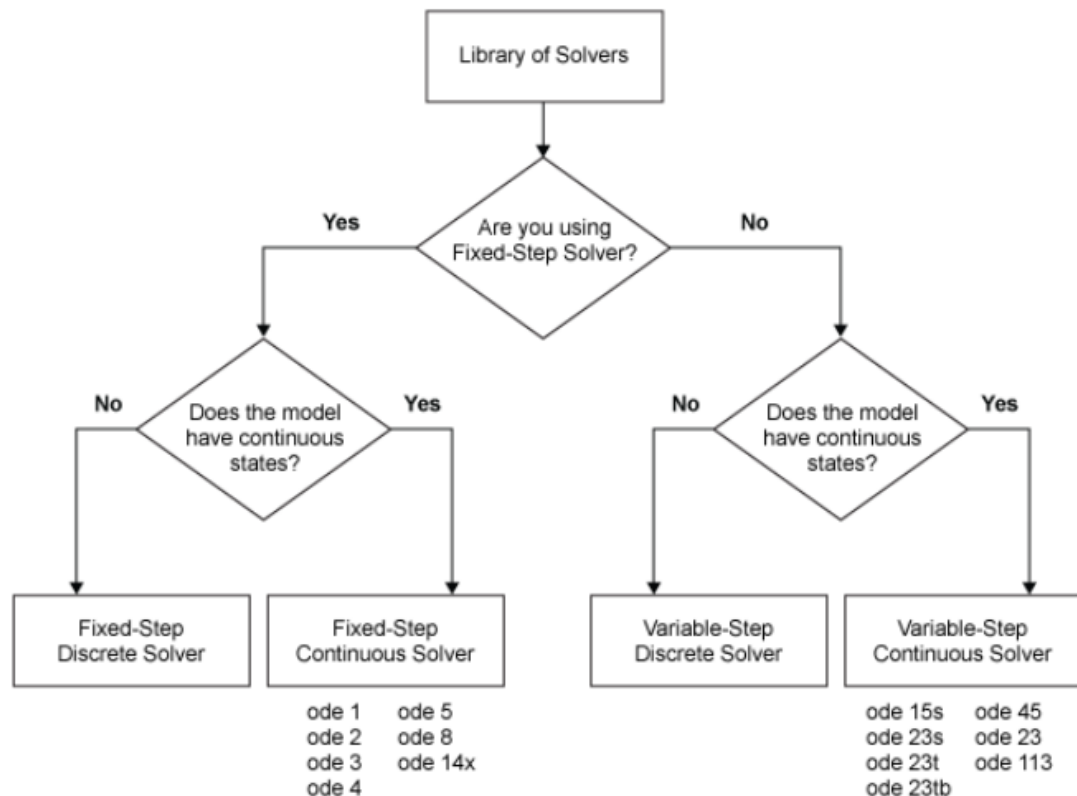
Simuloinnin asetuksissa on mahdollista määrittää simuloinnin kesto sekä simuloinnissa käytettävät asetukset. Käytettävien asetuksien avulla voidaan tasapainotella simuloinnin tuloksien tarkkuuden ja laskenta-ajan kanssa, sekä korjata epäjohtonmukaisia tuloksia tai epästabiilia toimintaa. Ratkaisijan valintaa voidaan arvioida kuvan 39 mukaisesti.



Kuva 38. Matlab-ratkaisijan asetukset.

Tässä työssä käytettiin muuttuvaa askeltyyppiä sekä sille asetettuja automaattisia arvoja. Muuttuvassa askeltyypissä askelten määrä määritellään automaattisesti simuloinnin sisällön mukaan, jolloin askelten määrää kasvatetaan, kun simulaatiossa tapahtuu suuria ja nopeita liikkeitä. Vastaavasti askelten määrää harvennetaan, kun liikkeet ovat tasaisia ja pieniä. Aikaisemmin luvussa 3.2.4 käytettäessä 3D-contact-toimintoa liikkeiden rajoittamiseen, huomattiin simulointiaikojen merkittävä kasvu. Tämä johtuu osakseen muuttuvan askeltyypin ratkaisijan käytöstä. Törmäystilanteessa simuloinnissa tapahtuu paljon nopeita liikkeitä, jolloin askelmäärää tihennetään automaattisesti simuloinnin tulosten tarkkuuden säilyttämiseksi.

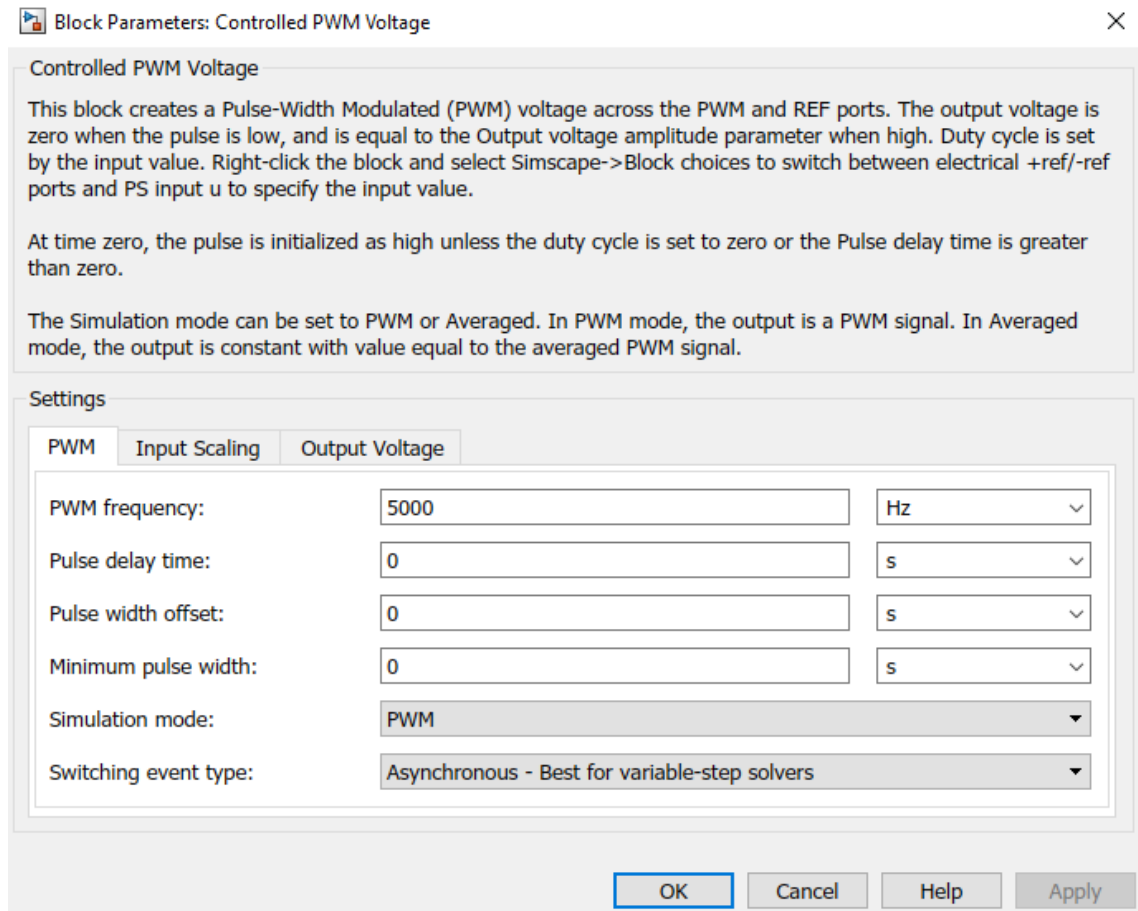
Ratkaisijana käytettiin ode45, Dormand-Prince -ratkaisijaa. Tämä ratkaisijamalli käyttää Runge-Kutta 4. ja 5. kertaluvun menetelmiä. Ode45 suositellaan käytettäväksi ensimmäisenä vaihtoehtona ja toimii useimpien ongelmien ratkaisemiseen. (Matlab documentation R2019b.) Epäjohdonmukaisten tulosten tai simuloinnin epästabiiliuden kohdalla ratkaisijamallin valintaa voi olla syytä arvioida uudelleen.



Kuva 39. Ratkaisijan valinnan arviointi.

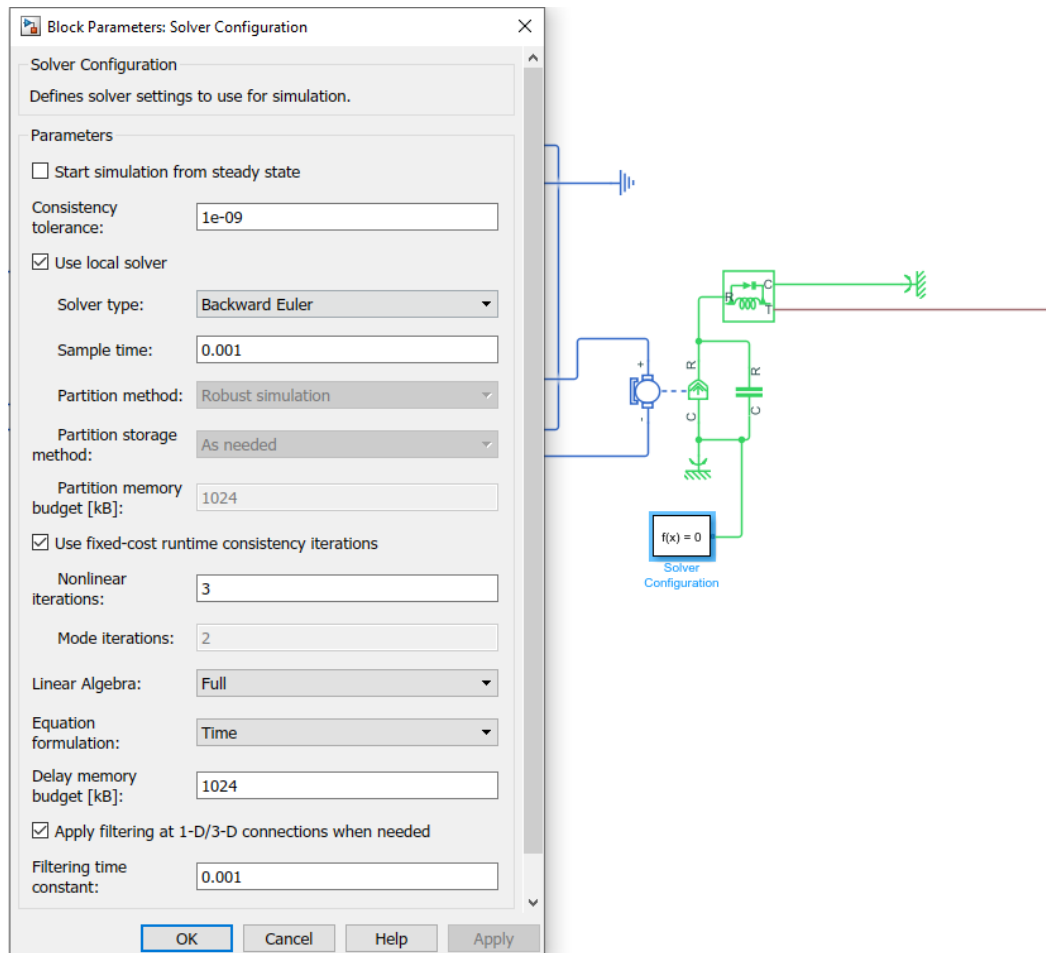
Erinäisten komponenttien toimintaa simulaation aikana ohjataan komponentin asetuksista. Esimerkiksi PWM-jännitelähteen simulaatioasetuksista on valittavissa PWM- ja averaged-asetusten välillä. PWM-jännitelähteen luonteesta johtuen simulointi saattaa osoittautua varsinkin korkeilla taajuuksilla raskaaksi. Tässä mallissa on käytetty PWM-asetusta, jolloin tietokone laskee jokaisen pulssin käyttäytymisen erikseen. Averaged-asetusta käyttäessä pulssien arvot keskiarvoistetaan, jolloin laskenta helpottuu. Tällöin simuloinnin tulosten tarkkuus saattaa kärsiä. Yleinen ohje on käyttää keskiarvoistettua menetelmää kar-

keiden tulosten saavuttamiseksi, jonka jälkeen tulokset tarkistetaan yksityiskohtaisemmalla menetelmällä.



Kuva 40. PWM-jännitelähteen asetukset.

Käytettäessä simscape-kirjaston komponentteja tulee käyttää aina paikallista ratkaisijaa. Paikallisten ratkaisijoiden avulla voidaan määritellä simscape-komponenteille erilainen ratkaisumalli ja näyteaika kuin muulle Simulink-mallille. Käyttämällä paikallista erillistä ratkaisijaa ja rajoittamalla iteraatioiden määrää saadaan simulaation nopeutta kasvatettua varsinkin tilanteissa, joissa tarkka yksityiskohtainen mallinnus ei ole tarpeen. Tällä on myös vaikutus tulosten tarkkuuteen.



Kuva 41. Paikallisen ratkaisijan käyttö simscape-komponenttien kanssa.

5 SIMSCAPE MULTIBODY ENVIRONMENT

Aikaisemmin läpikäydyn prosessin lisäksi päätettiin kevyesti tutkia erilaista lähestymistapaa digitaalisen kaksosen luontiin ja mallintamiseen. Vaihtoehtoisessa tavassa malli simuloidaan kokonaan Matlabin puolella käyttäen hyväksi Simscape Multibody Link-toiminnallisuutta. Se on Mathworksin oma plug-in Matlab R2008b ja sitä uudemmille versioille. Se on tuettu seuraavilla sovelluksilla:

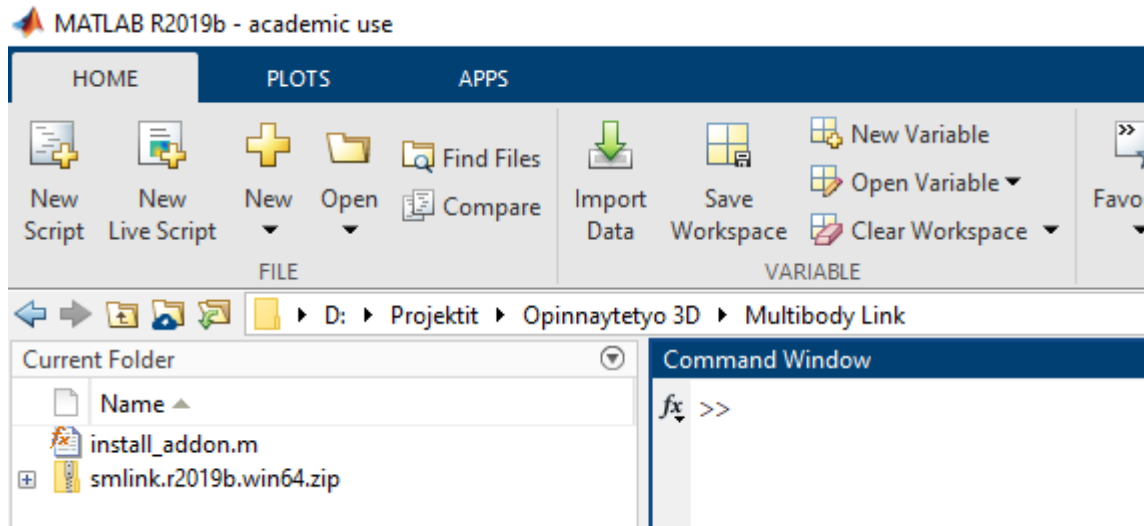
- SolidWorks 2001Plus eteenpäin
- Autodesk Inventor 2009 eteenpäin
- PTC Creo WildFire 2.0 eteenpäin, sekä Creo 1.0 eteenpäin

Toiminnallisuuden avulla yllä luetelluista sovelluksista voidaan lisäosan avulla tuoda CAD-malli Matlabin Multibody-ympäristöön. Lisäosa ei ole tätä opinnäytetyötä tehdessä tuettu vielä Siemens NX-ohjelmistolle. Toiminnallisuus on toteutettavissa muissakin CAD-ohjelmissa, mutta se vaatii oma-aloitteisuutta sekä ymmärrystä xml-ohjelmointikielestä. Tämän toiminnallisuuden tutustumisen kannalta oli mielekästä luoda aikaisemmin, kappaleessa 3, mallista kokoonpano ensin Autodesk Inventor-puolella Siemensin ohjelmiston asemesta.

5.1 Asennus

Tämä asennus käsittelee ainoastaan Autodeks Invetorin asennusta. Asennus aloitetaan lataamalla Simscape Multibody Link -lisäosa Mathworksin omilta kotisivuilta. Ladattavissa on kaikille Matlabin versioille oma versio. Näistä valitaan omaa asennettua Matlab-julkaisua vastaava tiedosto. Tiedosto on zip-muodossa ja se tallennetaan sellaiseen paikkaan, josta se on löydettävissä myöhemmin Matlabin puolella. Zip-tiedostoa ei pidä purkaa.

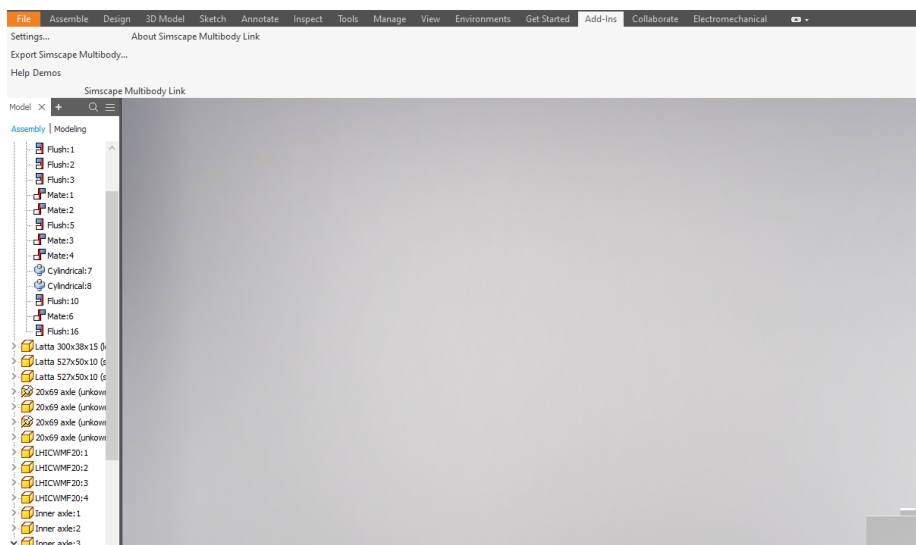
Tämän jälkeen Matlab käynnistetään järjestelmänvalvojan oikeuksilla, etsitään ladatut tiedostot Matlabin kansiorakenteesta ja suoritetaan komento "install_addon('smlink.r2019b.win64.zip')". Komentoa muokataan tarvittaessa kohdistamaan komento omaa asennettua ja ladattu lisäosaa vastaavaksi.



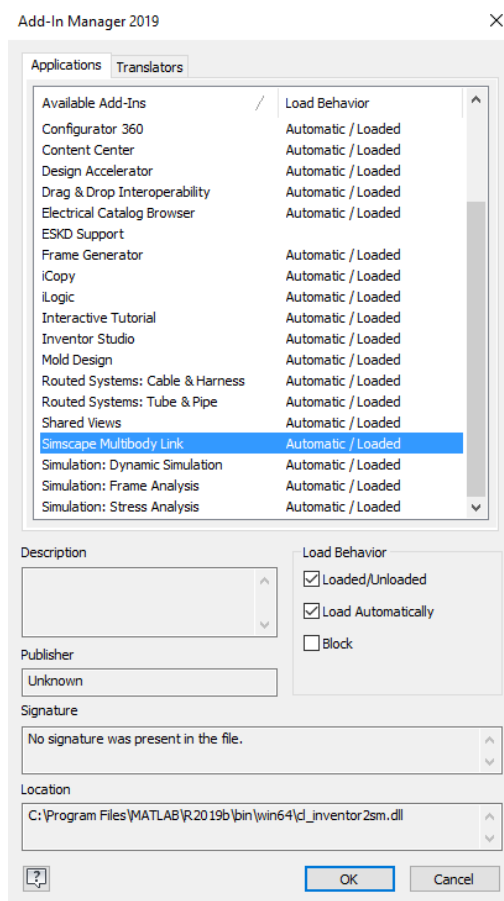
Kuva 42. Matlab kansiorakenteen valinta.

Asennuksen jälkeen rekisteröidään Matlab toimimaan automaattioserverinä lisäosalle kahden sovelluksen välistä keskustelua varten. Matlab komentoikkunassa suoritetaan ensin komento "regmatlabserver", jonka jälkeen windowsin komentopäätteessä komento "matlab -regserver".

Viimeisenä lisäosa tulee saattaa käytettäväksi suorittamalla Matlabin komentoikkunassa komento "smlink_linkinv". Tietokone käynnistetään uudelleen, jonka jälkeen lisäosa tulisi olla käytettävissä (kuva 43). Jos lisäosa ei ilmesty näkyviin, tulee se käydä Inventorin omista asetuksista laittamassa päälle. Kuvas-
sassa näkyvät laatikot (kuva 44).



Kuva 43. Autodesk Inventor lisäosien käyttö työn sisällä.



Kuva 44. Autodesk Inventor lisäosien hallintaikkuna.

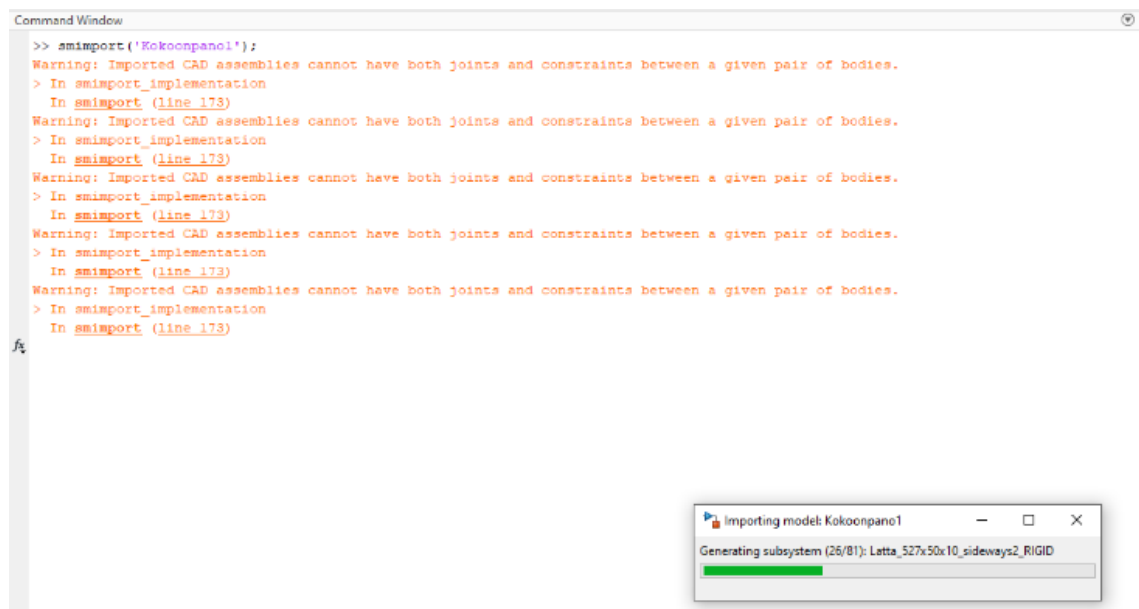
5.2 Mallin tuominen Multibody-ympäristöön

Malli voidaan viedä Inventorin puolelta Matlabin käsiteltäväksi valitsemalla Inventorin työkaluriviltä add-ins-välilehden "export simscape multibody"-komennolla

(kuva 43). Komento avaa ikkunan, jossa määritellään tallennuspolku. Tallennettavan tiedoston nimi ei saa sisältää erikoismerkkejä tai välilyöntejä, sillä Matlab ei yleisesti tunnista näitä.

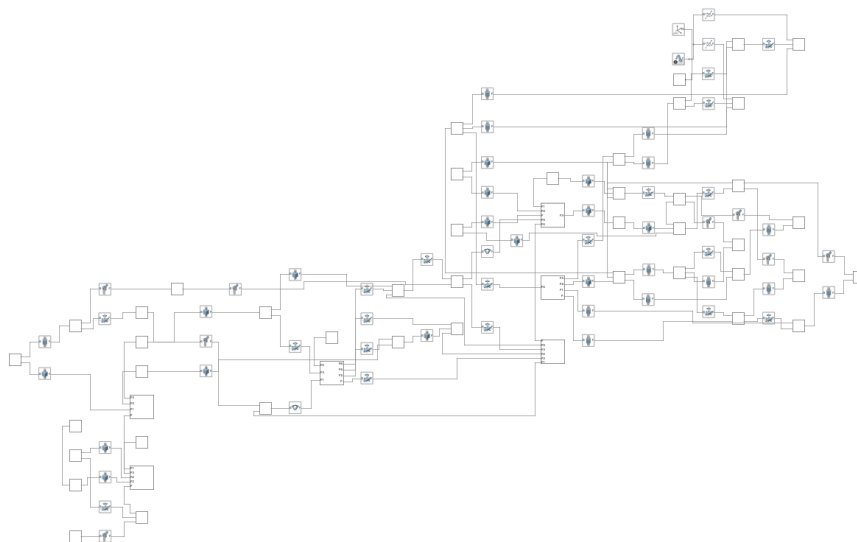
Viennin jälkeen Matlab avaa komentoikkunan, joka voidaan jättää huomioimatta. Malli avataan Matlabissa siirtymällä tallennettuun tiedostopolkuun Matlabin kansiorakenteessa kuvan 42 tavan mukaisesti ja suorittamalla komento "smimport('xxx');" ja vaihtamalla "xxx" oman, Inventorissa tallennetun, xml-tiedoston mukaisesti. (Matlab help center. Smimport-komennon käyttö.)

Matlab kääntää mallin simulink-malliksi, mikä saattaa tietokoneesta riippuen viedä aikaa. Tässä kohtaa Matlab-komentoikkuna täyttyy virhesanomista, jos malli on rakennettu Multibody-ympäristöä tukemattomalla tavalla (kuva 45). Virhesanomat tässä tilanteessa kertovat, että joitain kokoonpanon komponentteja on liitetty toisiinsa liian monilla rajoituksilla. Virhesanoma voi olla myös muun muassa materiaalitiedon puuttuminen, jolloin jollekin komponentille ei ole asetettu materiaalitietoa Inventorin puolella. Komponenttien rajoituksia karsittiin pois, jolloin virhesanomat poistuivat.

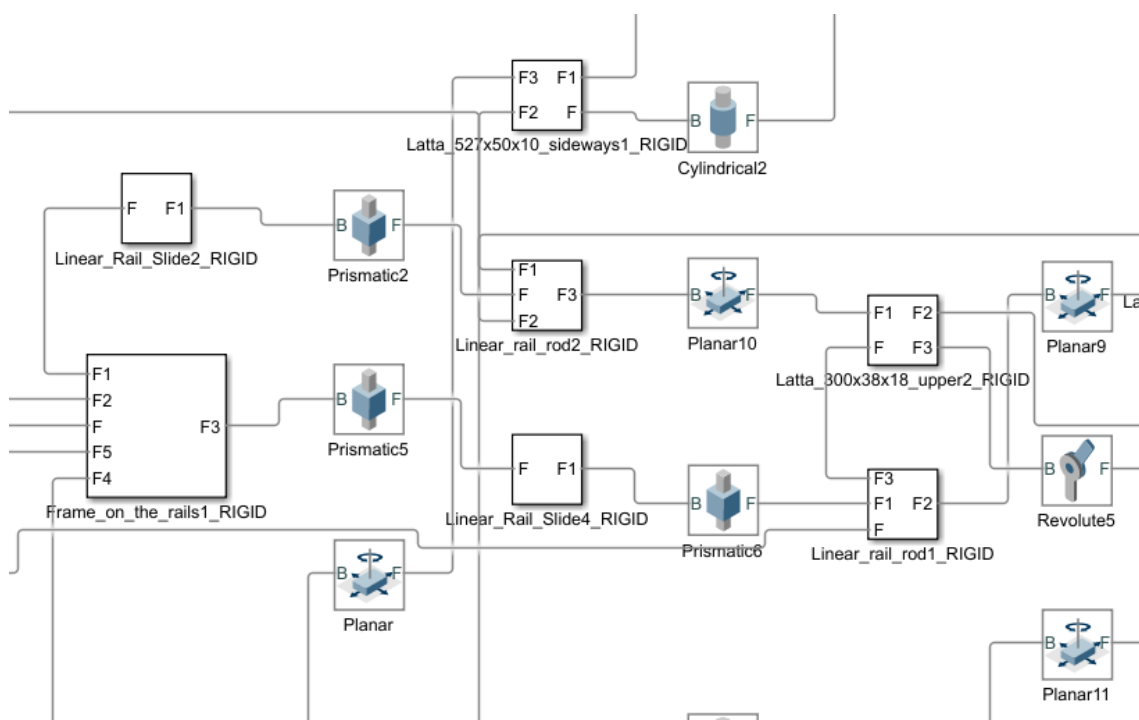


Kuva 45. Matlab komentoikkunan virhesanomiamia.

Kun vienti on valmis, avaa Matlab ikkunan, jossa kokoonpano on kokonaisuudessaan (kuva 46). Kokoonpanoa luodessa ja komponentteja piirrettäessä on kiinnitettävä huomiota komponenttien nimeämiseen, sillä niiden etsiminen tässä kohdassa voidaan todeta mahdottomaksi (kuva 47).



Kuva 46. Simscape multibody-malli digitaalisesta kaksosesta.



Kuva 47. Lähikuva Simscape multibody-mallista.

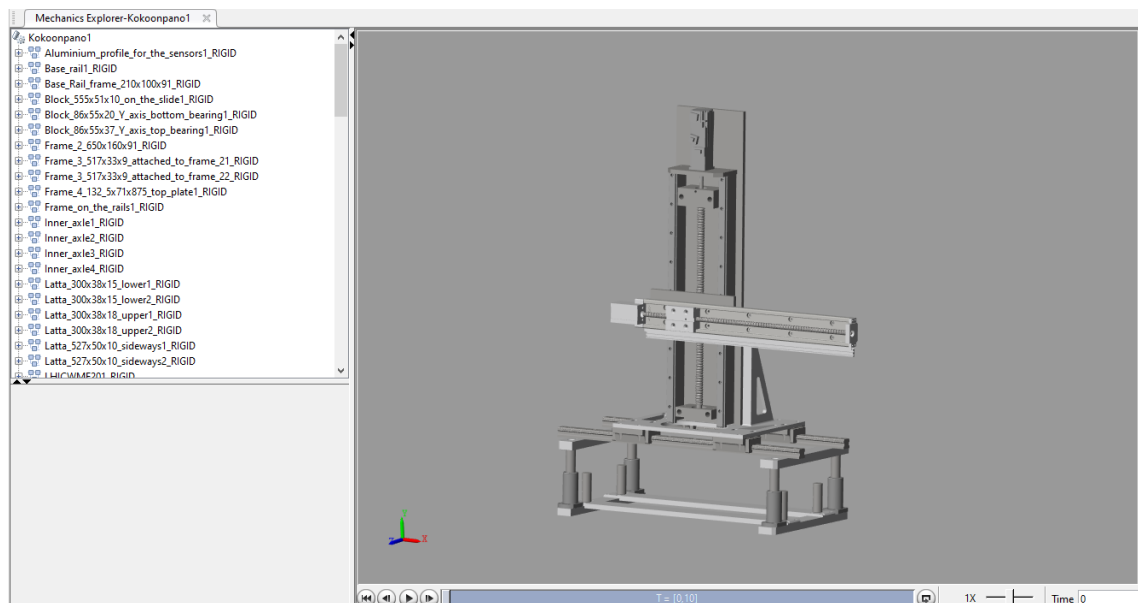
5.3 Simuloiminen

Kun luotu Multibody-malli on ehjä, on se simuloitavissa. Tässä mallissa huomattiin kuvan 48 mukaisia virheitä komponenttien tiedoissa, mitkä estivät simuloinnin suorittamisen. Virhesanoman mukaan komponentin massatieto ei ole johdonmukainen sen massakeskipisteen kanssa. Kyseiset virhesanomat koskivat SKF-läakerimalleja, jotka lopulta poistettiin kokoonpanosta kokonaan.



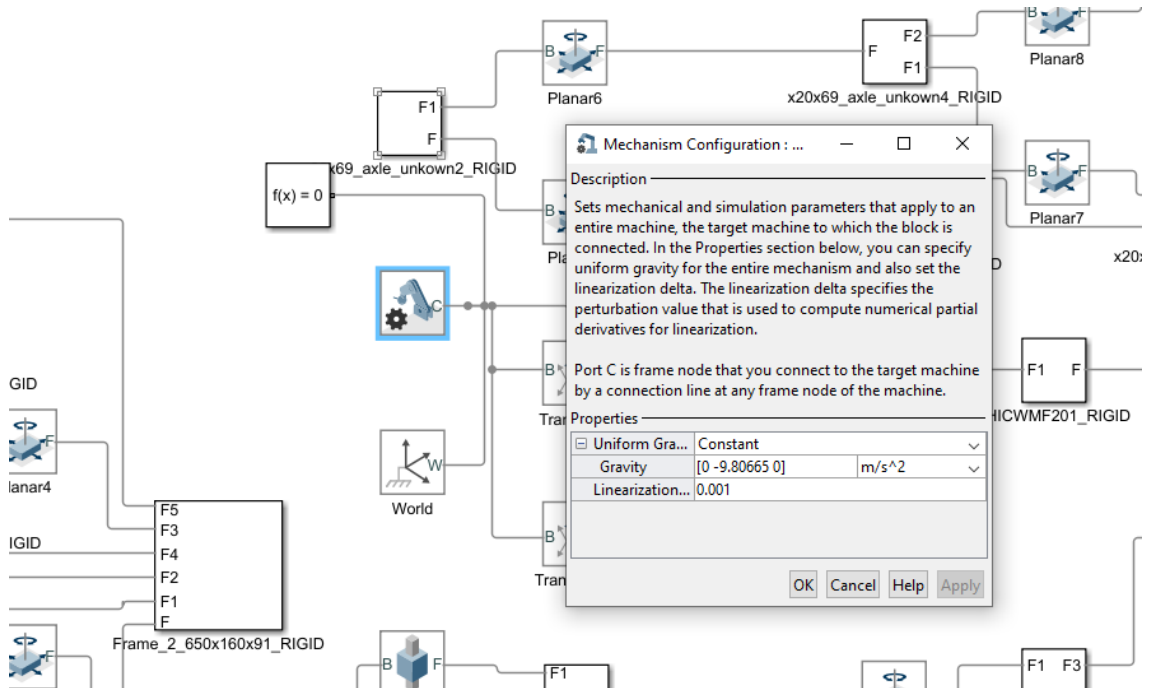
Kuva 48. Simulink-virhekonsoli.

Kun virheet ovat korjattu, voidaan mallia simuloida tavalliseen tapaan. Simulaation tuloksia (kuva 49) voidaan tarkastella suorittamalla Simulink-malli.



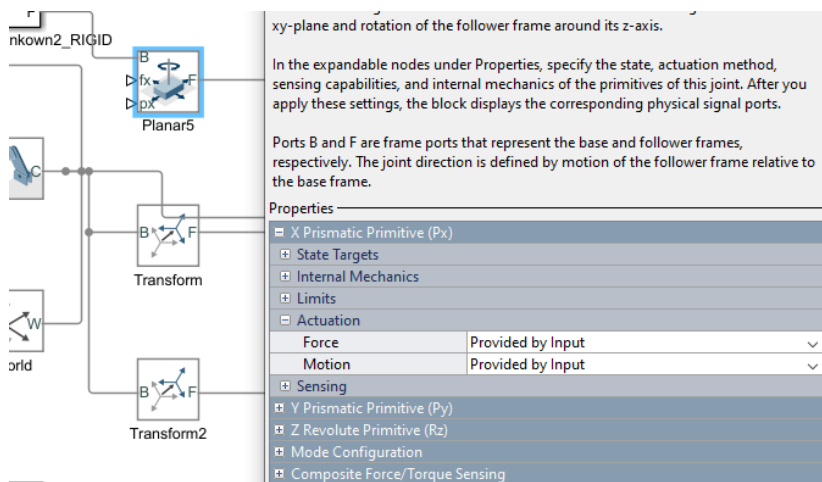
Kuva 49. Digitaalinen kaksonen Matlabin multibody-ympäristössä simuloituna.

Simulaation parametrejä voidaan määrittellä Simulink-mallissa (Kuva 50). Jokaisen komponentin takaa löytyy sille määriteltävissä olevia arvoja.



Kuva 50. Multibody-mallin painovoiman asetukset mallin takana.

Multibody-komponenttien asetusten takaa on mahdollista käydä laittamassa asetuksia päälle, jotka mahdollistavat ulkoisten voimien tuomisen komponenttiin (kuva 51). Pääperiaate on sama, kuin aikaisemmin käydyn Simulink-mallin kanssa.

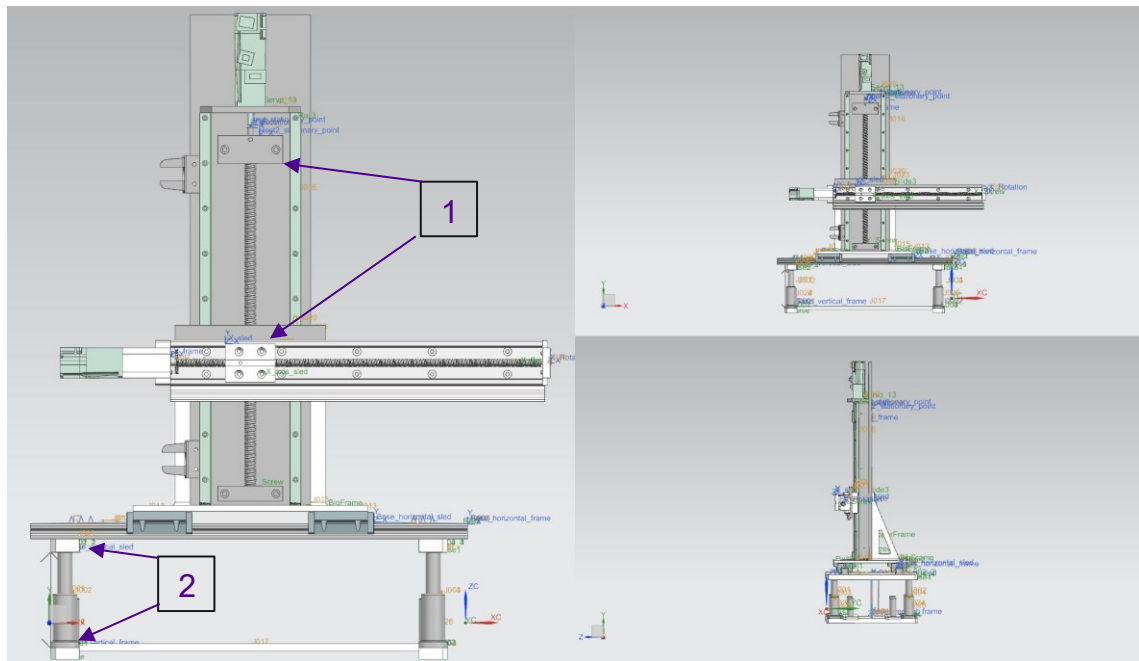


Kuva 51. Multibody-komponentin asetukset sisääntulojen mahdollistamiseksi.

6 TULOKSET

Tulosten tarkastelussa painotetaan huomio tulosten johdonmukaisuuteen ja kokonaisvaltaiseen toimintaan. Tämän opinnäytetyön puitteissa ei ollut mahdollisuutta lähteä kalibroimaan mallia oikean laitteen toiminnan mukaan, jolloin tulokset eivät sellaisinaan vastaa oikean laitteen toimintaa. Piirretyt komponentit ovat 1:1 malleja, jolloin mallin hienosäätö on toteutettavissa pääosin Matlabin puolella.

6.1 3D-malli



Kuva 52. Kokoonpano Siemens NX Motion-ympäristössä.

Kuvassa 52 esitetään 3D-mallin kokoonpano yleiskuva kolmesta eri kuvakulmasta. Mallissa on esitetty laatikoin etäisyysantureiden mittauspisteet:

1. Kelkan etäisyys mitataan ruuvijohteen yläpäästä kelkan rungon yläpäähän. Kelkan liikerata on 0 – 428 millimetriä.
2. Rungon etäisyys mitataan nuolien osoittamien osien reunoista. Mitattu etäisyys nollataan Matlabin Simulink-mallissa, jolloin ainoastaan rungon etäisyyden muutos huomioidaan. Muutos mitataan millimetreinä.

6.2 Simulaatio

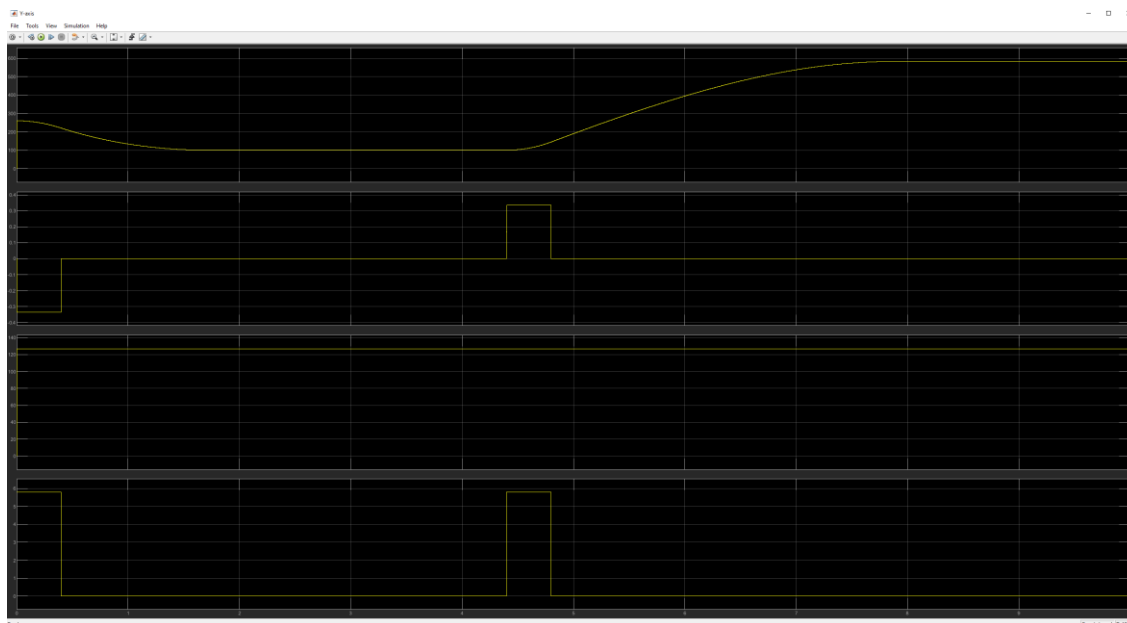
Tulokset esitetään kuvaajien avulla Matlab-ympäristössä. Kuvaajien avulla esitetään tulokset kolmessa eri käyttötilanteessa. Käyttötilanteista ensimmäisenä käsitellään laitteen toimintaa ilman PID-säädintä ajaen moottoria aikamääreellisen pulssin avulla rungon ollessa kiinteänä ilman jousia ja jousten kanssa. Toisena käyttötilanteessa moottoria ohjataan PID-säätimen avulla kiinteänä ja jousten kanssa. Viimeisenä tarkastellaan käyttötilannetta, jossa moottoria ohjataan PID-säätimen avulla samalla ottaen huomioon jousten värähtely moottorin paikoituksessa.

Jokainen kuvaaja koostuu neljästä eri mitattavasta suureesta ja ne ovat ylhäältä päin lueteltuna:

1. Kelkan etäisyys kelkan mittauspisteestä (kuva 52). Mitattava suure on etäisyys ja sen yksikkö on millimetri (mm).
2. Moottorin tuottama vääntömomentti. Mitattava suure on vääntömomentti ja sen yksikkö on newtonmetri (Nm).
3. Rungon etäisyyden muutos. Mitattava suure on etäisyys ja sen yksikkö on millimetri (mm).
4. Moottorin virta. Mitattava suure on virta ja sen yksikkö on ampeeri (A).

6.2.1 Pulssiohjaus

Kuvassa 53 esitetään tulokset ohjaukselle ilman PID-säädintä sekä kiinteällä rungonkiinnityksellä. Moottorille annetaan 0.4 sekunnin pulssi, jonka aikana moottori tuottaa sen nominaalista vääntömomenttiaan. Hieman ennen 5 sekunnin kohtaa moottorille annetaan samanpituinen, mutta vastakkaisen suuntainen pulssi, jonka aikana kelkka lähtee palaamaan kohti alkuperäistä asemaansa (kotiasema 260mm:n etäisyydessä).

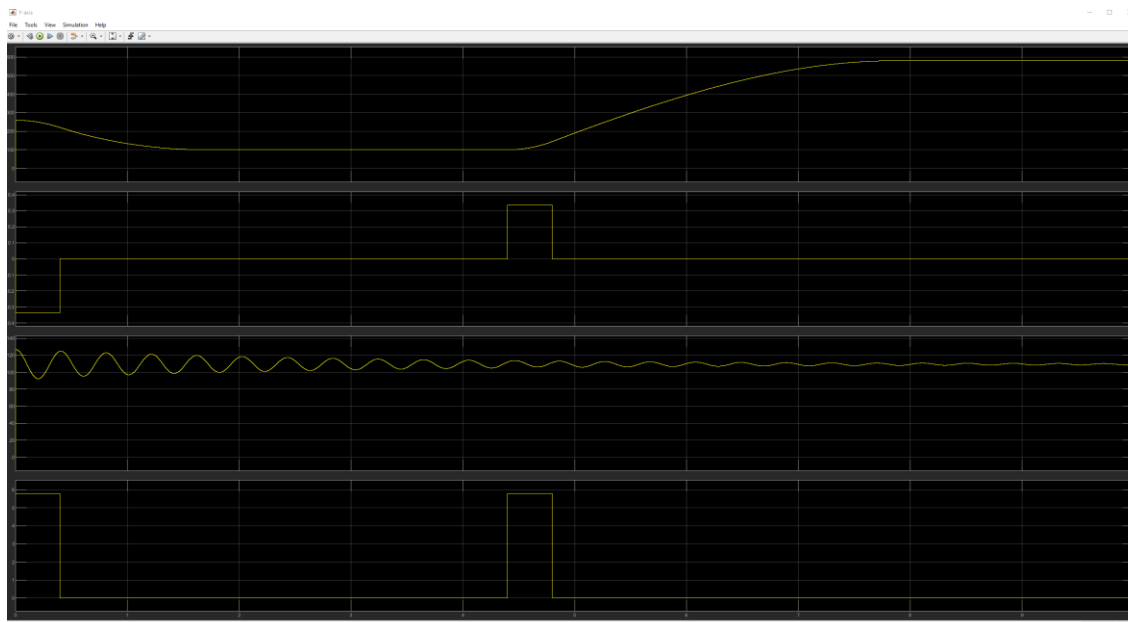


Kuva 53. Suora 0.4 sekunnin pulssiohjaus ilman takaisinkytkentää. Rungon ollessa kiinteä.

Moottorin saama virta on määritelty niin, että tuotettu momentti on lähempänä moottorin nominaalista vääntömomenttia, noin 0.37 newtonmetriä. Kelkka liikkuu 160 mm ensimmäisen pulssin aikana ylöspäin. Toisen pulssin aikana moottori liikkuu alaspäin reilusti yli lähtöasemansa 508 mm:iin. Ylitys on 320 mm alkupe-
räisestä asemastaan ja 108 mm laitteen käytännöllisestä toiminta-alueesta. Käytännössä kelkan käyttöalue on 0 – 428 millimetriä, jonka ylittyessä seurauksena on törmäys rungon ja kelkan välillä. Simulointiaikojen lyhentämiseksi törmäyskontaktit on otettu pois käytöstä (kts. luku 3.2.4).

Kelkan ajautuminen alaspäin mentäessä on selitettävissä painonvoiman vaikutuksesta. Ylös liikkua painovoima tekee työtä moottoria vastaan ja alas men-

täessä painovoima auttaa moottorin toimintaa. Ilman moottorin tuottamaa pysäyttävää vääntömomenttia kitkavoimat saavat kelkan pysäytettyä vasta huomattavasti myöhemmin 508 millimetrin kohdalla.

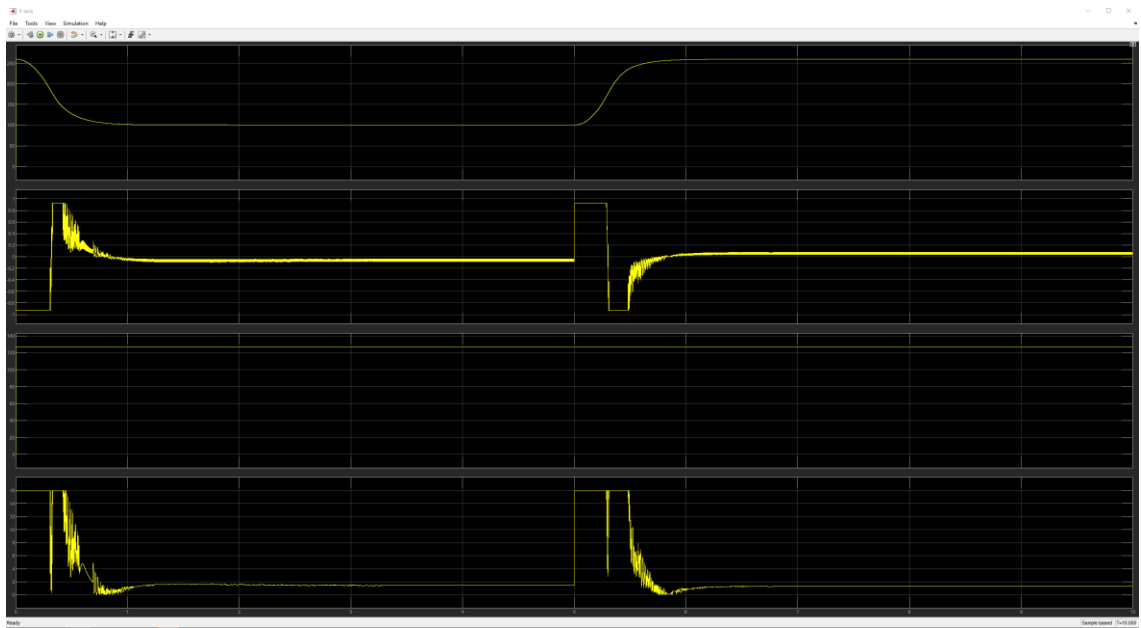


Kuva 54. Suora 0.4 sekunnin pulssiohjaus ilman takaisinkytkentää. Rungon ollessa ripustettuna jousiensä varassa.

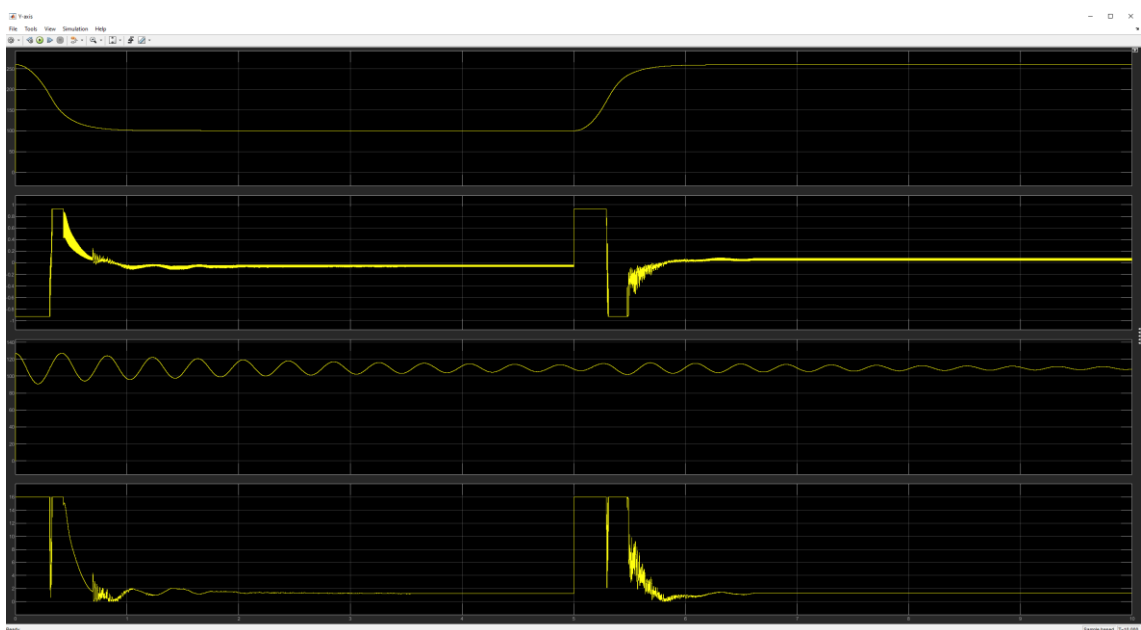
Kuvassa 54 esitetään samantyyppiset tulokset kuin kuvassa 53 muutoksena pelkästään rungon kiinnityksen vapautus. Kuvaajanelikon kolmannesta kuvajasta, ylhäältä päin, on nähtävissä kelkan rungon etäisyyden mittaus ja sen oskilloiva liike. Tämän moottorikäytön tuloksissa ei näy rungon oskillointi kelkan liikettä tarkasteltaessa, vaikka hypoteesi tämäntyyppisestä käytöksestä tehtiin. Tuloksien perusteella rungon aiheuttama värähtely ei ole tarpeeksi voimakasta vaikuttaakseen kelkan etäisyyden kehittymiseen, vaikka sellaista voitiin olettaa. Tuloksien tarkastelussa ja mittauksissa on kuitenkin huomioitava erinäiset mittausvirheet sekä kelkan hitausmomentti, jonka Siemens NX on laskenut ohjelman sisällä. Hitausmomenttia ja sen vaikutusta simulointiin ei ole tutkittu, jotta tulokset voitaisiin tarkemmin vahvistaa.

6.2.2 PID-säätö ilman värähtelyn kompensointia

Kuvassa 55 esitetään kelkan ohjaus käyttäen moottorin ohjaukseen PID-säätöä. Laitteen runko on kiinteästi kiinni, jolloin sen ei ole mahdollista joustaa kelkan liikkeen vaikutuksesta. Kelkkaa ajetaan ensin 160 millimetriä ylöspäin, jonka jälkeen kelkka palautetaan lähtöasemaansa, 260 millimetriin.



Kuva 55. PID-säädetty moottorinohjaus rungon ollessa kiinteä.



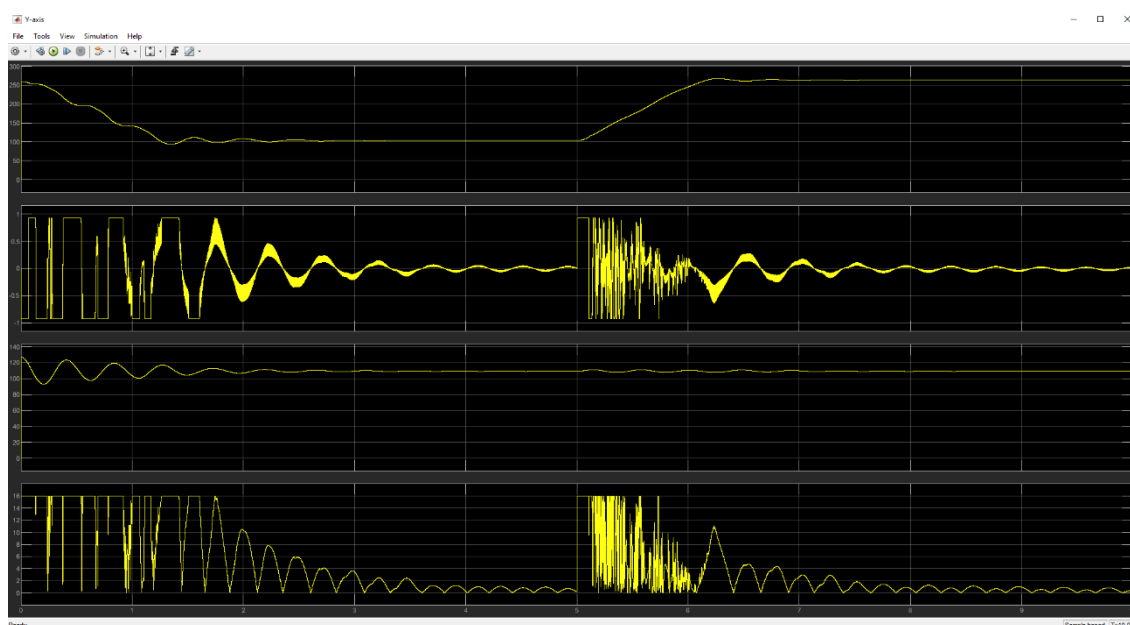
Kuva 56. PID-säädetty moottorinohjaus rungon ollessa ripustettuna jousien varassa.

Kelkan liike on tarkkaa eikä ylilyöntiä (*overshoot*) tapahdu. Moottorille ei ole valmistajan puolesta ilmoitettu maksimivääntö- ja nominaalisen vääntömomentin kuormitussuhdetta (*duty-cycle*), mutta kuvaajasta on luettavissa maksimaalisen vääntömomentin hetkittäinen tuotto, jonka jälkeen moottori palaa tuottamaan enemmän ja vähemmän sen jatkuvaa vääntömomentin arvoaan.

Kuvassa 56 esitetään samantyyppiset tulokset kuin kuvassa 55. Erona toimintaympäristössä on rungon vapaa liikkuminen sen jousiensa varassa. Tämä on nähtävissä moottorin tuottaman vääntömomentin sekä moottorin virtakuvaajan erilaisena käyttäytymisenä. Tässä moottorikäytössä kelkan värähtely aiheuttaa muutoksia moottorin toimintaan. Kelkan hitausmomentti aiheuttaa hetkittäisiä muutoksia moottorin kuormitukseen, joka ei selkeästi vaikuta kelkan liikkeeseen.

6.2.3 PID-säätö värähtelyn kompensoinnin kanssa

Kuvassa 57 on esitetty tulokset laitteen toiminnoille sellaisessa ympäristössä, jossa moottoria ohjataan PID-säätimen avulla, ja jossa runko kelluu vapaasti joustensa päällä aiheuttaen värähtelyä. Värähtelyä pyritään kompensoimaan PID-säätimen kanssa. Kelkka saavuttaa sille asetetut asemat hieman myöhemmin, kuin aikaisemmin.



Kuva 57. PID-säädetty moottorinohjaus värähtelyä kompensoiden rungon ollessa ripustettuna jousien päällä.

Rungon värähtelyn aiheuttama muuttuva kuormitus on nähtävissä kelkan oskiloivassa liikeradassa, etenkin liikkuesssa ensimmäisen liikkeen aikana ylöspäin. Rungon värähtelyn kompensointi vaimentaa rungön värähtelyä ja lopulta eliminoi sen melkein kokonaan. Alaspäin liikkuesssa kelkan aiheuttama värähtely on moottorille huomattavasti helpompi tehtävä, jolloin runko oskilloi ainoastaan vähän. Vertaamalla kuvien 56 ja 57 kuvaajanelikon kolmatta kuvaajaa voidaan todeta moottorikäytön avulla kompensoidun liikkeen konkreettiset erot.

7 POHDINTA

Työn päätavoitteena oli luoda digitaalinen kaksoiskappale TAMK:n automaatio-laboratoriossa sijaitsevasta laitteesta. Johdannossa kuvattiin myös työlle oleellisia osatavoitteita. Työ oli haastava ja omasta mielestäni tavoitteisiin päästiin hyvin. Simulaatiomallissa laitteen toisen akselin toimimattomuus jäi toistaiseksi mysteeriksi lukuisista korjausyrityksistä huolimatta. Vika johtunee luultavammin komponenttien liikkeiden vapausasteiden ristiriitaisuudesta. Mallin identifiointi jätettiin tämän työn ulkopuolelle, ja sen tutkiminen sekä suorittaminen on seuraava askel mallin kehittämiseen.

Opinnäytteen aihe oli työtä aloittaessa tuntematon, mutta erittäin mielenkiintoinen käsite. Tietoa kahden ohjelman yhteiskäytöstä simuloimiseen löytyi vähän ja eniten tietoa ohjelmien käytöstä tarjosi ohjelmien omat dokumentaatiot. Suoranaista kokemusta digitaalisen kaksosen luomisesta ei ollut, mutta tietokoneavusteisen suunnittelun perusteiden hallinta ja kokemus osoittautuivat arvokkaiksi aiheeseen perehtymisen ohella.

Tämän työn puitteissa tutustuttiin myös Simscape Multibody-ympäristöön tuomalla 3D-malli suoraan CAD-ohjelmasta Matlabin simuloitavaksi. Tämä tuo toisen näkökulman digitaalisen kaksosen simulointitapoihin ja mielestäni näiden kahden simulointimenetelmän rinnakkainen tutkiminen on aiheellista. Multibody-ympäristö vaikuttaa tuovan käyttäjälleen suoraviivaisemmat mahdollisuudet simuloinnin ympäristön ja rinnakkaistoimintojen määrittämiseen.

Digitaalisen kaksosen tuomat mahdollisuudet ovat rajattomat, tai ainakin ne tuntuvat siltä. Työtä tehdessä ja simulaatiota mallintaessa huomioitavien asioiden määrä tuntui välillä valtavalta. Simulaatiomallin reunaehtojen määrittäminen ennen mallintamisen aloittamista korostuu, ja jälkeinpäin pohdittuna tähän olisin kiinnittänyt enemmän huomiota. Prosessien mallinnus digitaalisesti tuottaa parhaimmillaan vähäisilläkin resursseilla arvokasta tietoa suunnittelijoilla, tuotannossa sekä loppukäyttäjälle.

LÄHTEET

About Mathworks. Luettu 2.5.2020. <https://se.mathworks.com/company/aboutus.html>

Bill Drury. 2001. Control techniques: Drives and controls handbook. Sivut 92-120.

Kangasrääsiö A. 2010. Adaptiivisen säädön menetelmät. Elektroniikan, tietoliikenteen ja automaation tiedekunta. Kandidaatintyö.

Matlab help center. Smimport-komennon käyttö. Luettu 10.4.2020 <https://se.mathworks.com/help/phymod/sm/ref/smimport.html>

FA Equipment for beginners (Servos). Mitsubishi electric. 2014. [https://www.mitsubishielectric.com/fa/assist/e-learning/pdf/eng/FA Equip for Begin eng Servos.pdf](https://www.mitsubishielectric.com/fa/assist/e-learning/pdf/eng/FA_Equip_for_Begin_eng_Servos.pdf)

Oguntoyinbo O. J. 2009. PID control or brushless DC motor and robot trajectory planning and simulation with Matlab/Simulink. Technology and communication. Vaasa university of applied sciences.

Ridal M. 2019. Digitaalisen kaksosen hyödyt teknologiateollisuudessa. Kone- ja tuotantotekniikan koulutus. Opinnäytetyö.

Siemens NX product website. Luettu 15.4.2020. <https://www.plm.automation.siemens.com/global/en/products/nx/>

Sustainability, 12, 1088; doi:10.3390/su12031088. 2020. Bevilacqua M., Bottani E., Ciarapica F. E., Costantino F., Donato L. D., Ferraro A., Mazzuto G., Monteriù A., Nardini G., Ortenzi M., Paroncini M., Pirozzi M., Prist M., Quatrini E., Tronci M., Vignali G.

Technical Explanation for Servomotors and Servo Drives. Omron guide. Luettu 26.3.2020 https://www.ia.omron.com/data_pdf/guide/14/servo_tg_e_1_1.pdf

Trevor English. 2017. Autodesk Inventor Blog. A Look at the History of Inventor Through the Eyes of Jay Tedeschi. Luettu 2.5.2020 <https://blogs.autodesk.com/inventor/2017/01/20/look-history-inventor-eyes-jay-tedeschi/>

Kuva 1, sivu 9. <https://i.pinimg.com/originals/bb/fb/c2/bbfbc2b4860824d08e6846f4971bb2b2.png>

Kuva 2, sivu 10. <https://assets.new.siemens.com/siemens/assets/api/uuid:c8de6dea-75eb-4f28-85c8-c8cfc596ad06/width:1125/crop:0:0:1:0,98786/quality:high/version:1563350638/digitaalinen-kaksonen-prosessiteollisuus.png>

Kuva 3, sivu 12. <https://www.ia.omron.com/support/guide/14/introduction.html>

Kuva 4, sivu 13. <https://realpars.com/wp-content/uploads/2019/06/Absolute-and-Incremental-Rotary-Encoder.png>

Kuva 4, sivu 14. By Arturo Urquizo - <https://commons.wikimedia.org/w/index.php?curid=17633925>

Kuva 6, sivu 15. https://www.sapctr.com/wp-content/uploads/MCAD_integration/Inventor/INVENTOR-integration-in-SAP-ECTR-1500x844.jpg

Kuva 7, sivu 16. <https://blogs.sw.siemens.com/wp-content/uploads/sites/2/2019/09/nx-12-1.jpg>

Kuva 8, sivu 16. https://upload.wikimedia.org/wikipedia/commons/e/e9/MAT-LAB_R2015b.png

LIITTEET

Liite 1. Matlab Simulink-malli.

